Version : **2020.01**

Dernière mise-à-jour : 2021/11/07 07:17

SER300 - Administration d'un serveur d'applications JEE avec Tomcat

Contenu du Module

Prérequis

Matériel

- Un poste (MacOS, Linux, Windows™ ou Solaris™),
- Clavier AZERTY FR ou QWERTY US,
- 4 Go de RAM minimum,
- Processeur 2 cœurs minimum,
- Un casque ou des écouteurs,
- Un micro (optionnel).

Logiciels

- Si Windows™ - Putty et WinSCP,
- Navigateur Web Chrome, Edge ou Firefox.

Internet

- Un accès à Internet **rapide** (4G minimum) **sans** passer par un proxy,
- Accès **débloqué** aux domaines suivants : https://ittraining.network, https://ittraining.io ainsi que leurs sous-domaines et https://rooms.ittraining.team
- Ports accessibles : 80,443.

Utilisation de l'Infrastructure

Au départ de votre formation, votre formateur vous attribue un ID allant de Trainee01 à Trainee10.

Pour avoir accès à vos machines virtuelles, vous devez d'abord vous connecter à votre gateway vers notre cloud. Ouvrez votre navigateur web **Chrome**, **Edge** ou **Firefox** et saisissez l'URL selon le tableau ci-dessous :

ID	URL (Notez http: et non https:)
Trainee01	http://compute01.ittraining.network
Trainee02	http://compute02.ittraining.network
Trainee03	http://compute03.ittraining.network
Trainee04	http://compute04.ittraining.network
Trainee05	http://compute05.ittraining.network
Trainee06	http://compute06.ittraining.network
Trainee07	http://compute07.ittraining.network
Trainee08	http://compute08.ittraining.network
Trainee09	http://compute09.ittraining.network
Trainee10	http://compute10.ittraining.network

Dans la boîte de connexion, entrez votre ID et le mot de passe qui vous a été **fourni par votre formateur**.

Cliquez ensuite sur la connexion **Gateway-XX_SSH** ou XX est le numéro dans votre **ID**.

Si vous souhaitez avoir accès à votre Gateway directement en utilisant une connexion SSH, utilisez la commande appropriée issue du tableau suivant :

ID	Commande
Trainee01	ssh -l trainee compute01.ittraining.network -p 21022
Trainee02	ssh -l trainee compute02.ittraining.network -p 21122
Trainee03	ssh -l trainee compute03.ittraining.network -p 21222
Trainee04	ssh -l trainee compute04.ittraining.network -p 21322
Trainee05	ssh -l trainee compute05.ittraining.network -p 21422
Trainee06	ssh -l trainee compute06.ittraining.network -p 21522
Trainee07	ssh -l trainee compute07.ittraining.network -p 21622
Trainee08	ssh -l trainee compute08.ittraining.network -p 21722
Trainee09	ssh -l trainee compute09.ittraining.network -p 21822
Trainee10	ssh -l trainee compute10.ittraining.network -p 21922

Utilisez le mot de passe qui vous a été fourni par votre formateur.

L'adresse IP de la machine virtuelle est :

Machine	Nom d'hôte	Adresse IP
CentOS7	centos7.i2tch.loc	10.0.2.51

Les noms d'utilisateurs et les mots de passe sont :

Utilisateur	Mot de Passe
trainee	trainee
root	fenestros

Dernièrement connectez-vous à la machine virtuelle utilisée pour cette formation :

```
$ ssh -l trainee 10.0.2.51
```

Programme de la Formation

Jour #1

- **SER300 - Administration d'un serveur d'applications JEE avec Tomcat** - 1 heure.
  - Contenu du Module
  - Prérequis
    - Matériel
    - Logiciels
    - Internet
  - Utilisation de l'Infrastructure
  - Programme de la Formation
- **SER301 - Présentation des Technologies** - 2 heures.
  - Présentation de Tomcat 8
    - Historique et différentes versions
  - Rappel sur les applications Web en Java
  - Contenu statique, dynamique, Servlets, JSPs et Composants EJB
    - Servlets
    - JSP
    - Enterprise JavaBeans - EJB
  - Le Modèle MVC
  - Les Modules Java EE
    - Modules Web
    - Modules EJB
    - Modules Clients
    - Modules de Connecteurs
  - Positionnement d'Apache Tomcat dans la norme Java EE
    - Structure d'une Application Web
    - Le Descripteur de Déploiement web.xml

        - Les Sessions HTTP

- **SER302 - Installation de Tomcat 8 et les serveurs associés** - 2 heures.
    - Désactiver SELinux
    - Tomcat et JDK
    - Apache
        - Présentation d'Apache
        - Installation
        - Testez le serveur apache avec telnet
    - Coupler Tomcat et Apache
    - MariaDB
        - Présentation
        - Installation
        - Configuration
    - OpenLDAP
        - Présentation
        - Installation

- **SER303 - Configuration du serveur Tomcat 8** - 2 heures.
    - Architecture du Serveur
    - Fichiers de Configuration
        - Le Fichier $CATALINA_HOME/conf/server.xml
            - L'élément <Server>
            - L'élément <Service>
            - L'élément <Connector>
            - L'élément <Executor>
            - L'élément <Engine>
            - L'élément <Host>
            - L'élément <Context>
            - L'élément <Realm>
            - L'élément <Loader>
            - L'élément <Manager>
            - L'élément <Store>
            - L'élément <Valve>
                - Filtrage de l'adresse IP

        - Filtrage de nom de la machine du client
        - LAB #1 -Journalisation des Requêtes Client dans un Fichier Texte
        - LAB #2 -Journalisation des Requêtes Client dans une Base de Données
      - L'élément <Listener>
    - Le Fichier $CATALINA_HOME/conf/web.xml
    - Le Fichier $CATALINA_HOME/conf/tomcat-users.xml
    - Le Fichier $CATALINA_HOME/conf/catalina.policy
    - Configuration des Ressources
      - Portée des Ressources
      - Pools de Connexion
      - Sessions JavaMail
      - JavaBeans
      - Entrées D'Environnement

Jour #2

- **SER304 - Déploiement et Gestion des Applications** - 3 heures.
  - Déployer une application
  - Déploiement Automatique
  - L'Élément Context
  - Déploiement avec XML
  - Application Manager de Tomcat
    - L'interface Texte
      - list
      - deploy
      - start
      - stop
      - reload
      - undeploy
      - resources
      - serverinfo
    - L'interface HTML
    - L'interface ANT

- Deployer de Tomcat

- **SER305 - Sécurité du serveur Tomcat 8** - 4 heures.
  - Authentification, Autorisation et Cryptage
    - Authentification
    - Autorisation
    - Cryptage
  - La Sécurité sous Tomcat
  - Configuration
    - Realms
      - User Database Realm
      - JDBC Realm
      - DataSource Realm
      - JNDI Realm
        - Le format LDIF
          - La commande ldapadd
      - JAAS Realm
      - Combined Realm
      - LockOut Realm
    - Tomcat et le SSO
    - Tomcat et le SSL
      - Présentation de SSL
        - Fonctionnement de SSL
      - Configurer Tomcat
      - Configurer Apache
        - Installation de SSL
        - Configuration de SSL
        - Mise en place des paramètres de sécurité SSL
        - Tester Votre Configuration
      - Apache en Frontal HTTPS
      - Restrictions d'Accès
      - Le Gestionnaire de Sécurité

Jour #3

- **SER306 - Journalisation, Supervision et Clustering** - 6 heures.
  - Configuration des journaux
    - java.util.logging
    - log4j
  - Supervision
    - JMeter
    - Interface JMX
    - JConsole
  - Clustering avec Tomcat
    - Préparation
    - Le Cluster de Répartition de Charge avec Apache et mod_jk
    - Le Cluster de Répartition de Charge avec Apache et mod_proxy_ajp
    - Le Cluster en mode Maître/Esclave
    - Maintenir l'Etat des Clients
      - Préparation
      - Sessions Persistantes sur Système de Fichiers
- **SER307 - Validation de la Formation** - 1 heure.
  - Pour Aller Plus Loin
    - Support de Cours
    - L'Infrastructure Hors Formation
      - Matériel
      - Logiciels
      - Machine Virtuelle
  - Rappel du Programme de la Formation
    - Jour #1
    - Jour #2
    - Jour #3
  - Remettre en Etat l'Infrastructure
  - Évaluation de la Formation
  - Évaluation des Acquis
  - Remerciements

SER301 - Présentation des Technologies

Contenu du Module

Présentation de Tomcat 8

Historique et différentes versions

Historiquement le serveur d'applications Java **Jakarta Tomcat** appartenait à la première catégorie du projet **Jakarta** issu de la **Fondation Apache**.

Le projet Jakarta est est divisé en trois catégories liées aux technologies **Java** :

- Les serveurs d'application,
- Les bibliothèques, outils et API,
- Les frameworks.

Tomcat a ses origines dans les **Conteneurs Web**, aussi appelés *Moteur de Servlet*, de SUN Microsystems et de la fondation Apache :

- Java Web Server
- JServ

En **1999**, SUN Microsystems donne le code de Java Web Server à la fondation Apache. Tomcat est né de la fusion du projet JServ avec ce dernier.

Tomcat est devenu un projet important et de ce fait a trouvé sa propre place au sein de la Fondation Apache. Quittant le projet Jakarta, Tomcat est maintenant appelé **Apache Tomcat**.

Les différentes versions de Tomcat sont :

Version Tomcat	API Servlet	API JavaServer Pages	Spécification Java JEE
3.x	2.2	1.1	J2EE 1.2
4.x	2.3	1.2	J2EE 1.3
5.x	2.4	2.0	J2EE 1.4
6.x	2.5	2.1	Java EE 5 et +
7.x	3.0	2.2	Java EE 6 et +
8.x	3.1	2.3	Java EE 7 et +
9.x	4.0	2.3	Java EE 8 et +
10.0.11	5.0	3.0	Java EE 8 et +

Rappel sur les applications Web en Java

Java prend ses origines dans le développement d'applications pour des terminaux mobiles. Débuté en **1991**, le projet **Star 7** a d'abord été basé sur l'utilisation du langage C++. Cependant ce langage, à cause de la gestion mémoire contraignante, a été mis de côté et remplacer par un langage nouveau le **C++–**. Ce langage a été ensuite connu sous le nom **OAK** pour ensuite devenir **Java**.

En **1995** la première version du **JDK** (*Java Development Kit*) a été rendu disponible. Il permettait de développer :

- des applications graphiques,
- des applications client/serveur,
- des applets (applications embarquées dans des pages HTML).

Lors du développement d'une application, la compilation du code source Java donne un format de fichier spécifique appelé **byte-code**. Ce format de fichier est interprété par une **machine virtuelle java**.

Java se décompose en plusieurs plate-formes :

- JSE (*Java Standard Edition*),
  - une plate-forme de base pour le développement d'applications clients/serveurs et graphiques ainsi que des applets qui est en deux formes :
    - JDK (*Java Development Kit*),
      - nécessaire pour le développement sous Java,
    - JRE (*Java Runtime Environment*),
      - exécute des applications java,
- JEE (*Java Enterprise Edition*),
  - une extension de JSE qui permet le développement d'applications qui s'exécutent sur un serveur d'application et qui peuvent être exploitées par :
    - des clients légers, tels des navigateurs web,
    - des clients lourds, tels des applications graphiques fenêtrées.
- JME (*Java Micro Edition*),
  - une plate-forme de développement d'applications mobiles utilisées dans des téléphones mobiles, pocket pc etc.

En **1998**, la version 1.2 de ces plate-formes, connue sous le nom de **Java 2** a donné naissance à l'utilisation des termes **J2SE**, **J2EE** et **J2ME**.

En **2004**, la version 1.5 a été publiée, donnant lieu à **Java 5**.

En **2006**, la version 1.6 ou **Java 6** a été publiée et le chiffre 2 retiré de J2SE et J2EE.

La version **actuelle** de la plate-forme est la version **8**.

Le JRE de la plate-forme JSE est constitué des éléments suivants :

- **JVM** (*Java Virtual Machine*),
  - Il existe une JVM pour la majorité des systèmes d'exploitation, rendant ainsi portables les applications Java,
  - La JVM gère directement la mémoire des applications,
- **Bibliothèque de classe Java**,
  - Des composants logiciels prêt à l'emploi,

Outre les deux éléments précédents, le JDK de la plate-forme JSE contient des **outils de développement** suivants :

- **javac**,
  - un compilateur de code source Java,
- **java**,
  - un interpréteur,
- **javadoc**,
  - un générateur de documentation.

Contenu statique, dynamique, Servlets, JSPs et Composants EJB

Le modèle de développement JEE contient trois types de composants logiciels.

Servlets

Le rôle d'une servlet dans une application est :

- de recevoir les requêtes des clients,
- d'en extraire des informations,
- de formater ces informations,
- de préparer des données nécessaires à la génération d'une réponse.

Les servlets sont :

- des composants logiciels écrits en Java,
- des composants orientés requête/réponse,

- portables et évolutives,
- chargées en mémoire lors de leurs premier appel,
- déchargées de la mémoire lors de l'arrêt de l'application ou du serveur.

Il n'y a qu'une seule instance d'une servlet en mémoire. Le serveur utilise ensuite un **thread** pour traiter chaque requête.

La servlet est une classe Java qui a un cycle de vie spécifique. La servlet est :

- **instancier** et **charger** en mémoire,
- **initialiser** grâce à l'appel de la méthode **init()** par le serveur,
- **prête** - les requêtes sont satisfaites par la méthode **service()**,
- **détruite** par le serveur en appelant la méthode **destroy()**,
- **nettoyée** par la machine virtuelle Java.

JSP

Proches des pages PHP et ASP, les **JavaServer Pages** permettent le développement de contenu mélangé statique et dynamique.

Le rôle d'une page JSP dans une application est de prendre en charge la partie visuelle de l'application en présentant des données au client.

Une page JSP :

- possède une extension **.jsp**,
- est un squelette de code HTML pour la partie statique,
- contient du code Java pour l'intégration des données dynamiques,
- doit être traitée par le serveur avant d'être renvoyée au client.

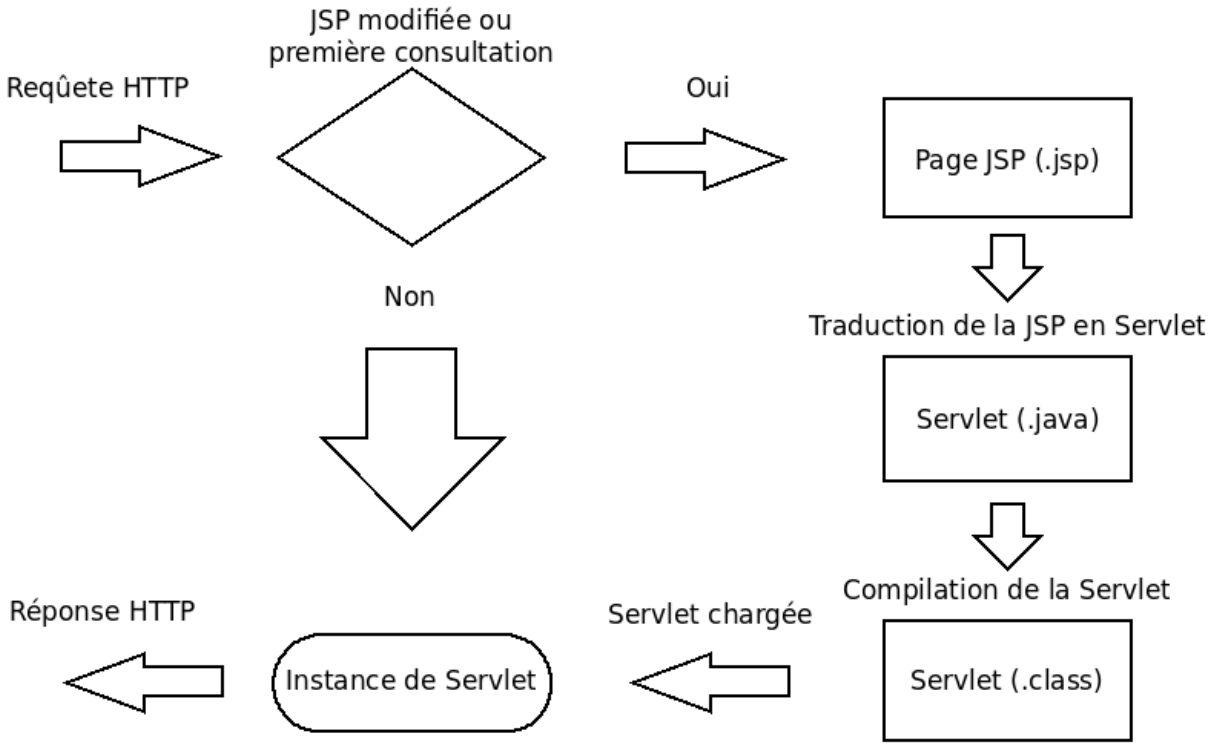
Le traitement est effectué au premier appel de la page et à chaque fois que la page est modifiée par un programmeur. C'est pour cette raison que le serveur a besoin d'un JDK car il transforme la JSP en classe Java puis la compile en servlet.

Une requête vers une page JSP peut être résumée ainsi :

- l'arrivée de la première requête,
- la transformation de la page .jsp en code source java (**.java**),

- la compilation du code source en classe Java (**.class**) qui devient ainsi une servlet,
- le départ de la première réponse,
- l'arrivée d'une deuxième requête,
- la délégation de la requête par le serveur *directement* à la servlet précédemment créée,
- le départ de la deuxième réponse,
- etc ...

Le schéma ci-dessous résume ce mécanisme :



Voici un exemple simplifié d'une page jsp :

```
<HTML>
    <HEAD>
        <TITLE>JSP Page Example</TITLE>
```

```
    </HEAD>
    <BODY>
        This is a random number:
        <%= Math.random() %>
    </BODY>
</HTML>
```

Enterprise JavaBeans - EJB

Les EJB sont des composants métiers distribués. Il existe deux types d'EJB :

- EJB Session,
- EJB piloté par message, appelé **MDB** (*Message Driven Bean*).

Important - Notez que les **EJB Entités** n'existent plus et sont aujourd'hui remplacés par la nouvelle API de persistance Java : **JPA** (*Java Persistence API*). Les EJB sont hébergés dans une partie spécifique d'un serveur d'applications Java EE. Apache Tomcat ne disposant pas de cet environnement n'utilise pas les EJB.

Le Modèle MVC

La plate-forme Java EE comporte un cycle de conception d'applications :

- le développement des composants applicatifs,
- l'assemblage des composants en modules,
- l'assemblage des modules en applications,
- le déploiement de l'application.

Afin de suivre ce cycle, le développement d'applications utilise le **modèle MVC** (*Model View Controller*) qui :

- a ses origines dans le langage **SmallTalk** au début des années 1980,
- divise l'application en trois parties distinctes :
    - le **modèle**,

    - la **vue**,
    - le **contrôleur**.

Ce modèle, appliqué au développement dans la plate-forme JEE, correspond aux composants logiciels suivants :

- le modèle = **EJB/JPA**,
- la vue = **JSP**,
- le controleur = **servlets**.

Lors d'une requête d'un client vers l'application ainsi développée on constate que :

- la requête http est reçue par la servlet qui extrait des informations,
- ces informations sont utilisées pour appeler des composants EJB/JPA adéquats,
- les composants EJB/JPA manipulent les données (lecture, enregistrement, mise à jour ...),
- les composants EJB/JPA retournent les données ainsi modifiées à la servlet,
- la servlet appelle la JSP adéquate,
- la JSP inclut les données dans la génération de la réponse au client.

Pour simplifier le développement en utilisant le modèle MVC, certains outils sont disponibles :

- **Apache Struts**,
- **JavaServer Faces**.

Les Modules Java EE

Les différentes ressources d'une application sont organisées en 4 **modules** selon le rôle qu'elles jouent :

- modules pour l'interface web,
- modules pour les clients lourds,
- etc ...

Modules Web

Ces modules intègrent :

- des servlets,
- des JSP,
- des pages statiques en html,
- des bibliothèques de classes Java ayant une extension **.jar**.

Ces modules possèdent un **descripteur de déploiement** sous la forme d'un fichier appelé **web.xml** et sont assemblés dans un fichier compressé au format **zip** ayant une extension **.war**.

Modules EJB

Ces modules intègrent :

- des fichiers de code Java,
- des fichiers de configuration spécifiques à un serveur donné.

Ces modules possèdent un **descripteur de déploiement** sous la forme d'un fichier appelé **ejb-jar.xml** et sont assemblés dans un fichier compressé au format **zip** ayant une extension **.jar**.

Modules Clients

Ces modules permettent l'utilisation des EJB à travers un client lourd ayant été développé avec un API de programmation tel **AWT** ou **SWING**.

Ces modules possèdent un **descripteur de déploiement** sous la forme d'un fichier appelé **application-client.xml** et sont assemblés dans un fichier compressé au format **zip** ayant une extension **.jar**.

Modules de Connecteurs

Ces modules permettent l'accès aux données externes en utilisant des API tels **JDBC**, **JNDI** ou encore **JCA**.

Ces modules possèdent un **descripteur de déploiement** sous la forme d'un fichier appelé **ra.xml** et sont assemblés dans un fichier compressé au format **zip** ayant une extension **.rar**.

Positionnement d'Apache Tomcat dans la norme Java EE

Le serveur Tomcat ne dispose pas de l'environnement spécifique nécessaire pour héberger les EJB. Par conséquent il n'est pas possible d'utiliser les EJB avec Tomcat. En effet, des modules cités précédemment, seuls les modules web peuvent être exploités par un serveur Tomcat.

Structure d'une Application Web

Chaque application web est stockée dans un répertoire distinct dans le répertoire **/usr/tomcatX/webapps/** où X représente la version de Tomcat.

Le répertoire de l'application est organisé de la façon suivante :

```
|-- MonApplication
|   |-- WEB-INF
|   |   |-- classes
|   |   |-- lib
|   |   `-- web.xml
|   |-- images
|   |-- page.jsp
|   `-- public.html
```

Le répertoire **WEB-INF** est la partie privée de l'application tandis que les fichiers stockés en dessus du répertoire **MonApplication** constituent la partie publique.

Dans le répertoire WEB-INF se trouve :

- un répertoire **classes** pour stocker les classes Java,
- un répertoire **lib** pour stocker les bibliothèques de code Java,
- le descripteur de déploiement **web.xml**.

Chaque application web est accessible par une URL unique composée :

- du nom d'hôte du serveur,
- d'un numéro de port spécifique,
- du chemin du **contexte d'application web** qui représente une vue globale de l'application.

L'URL prend donc la forme **http://nom_d_hote:port/contexte**.

Le Descripteur de Déploiement web.xml

Le **Descripteur de Déploiement** a pour but d'aider le serveur à installer l'application qu'il décrit. Il peut contenir 5 informations principales :

- des paramètres d'initialisation de l'application ou des servlets,
  - informations de type texte consultées par les servlets et JSP de l'application, par exemple, l'adresse email de l'administrateur de l'application,
- la définition des servlets et des JSP,
  - la déclaration de chaque servlet est necéssaire afin qu'elle puisse être utilisée. Les pages JSP doivent aussi être déclarées si elles ont besoin de paramètres d'initialisation particuliers,
- la correspondance des servlets avec des URLs,
  - les classes Java des servlets se trouvent dans la partie privée de l'application : **WEB-INF/classes**. Le serveur d'applications doit associer un URL avec chaque servlet,
- les détails des pages d'accueil et d'erreur de l'application,
  - définition de la page d'acceuil de l'application ainsi que la correspondance entre des erreurs HTTP spécifiques et les pages HTML à afficher,
- des contraintes de sécurité,
  - informations concernant quels utilisateurs ont accès à quelles ressources ainsi que comment ces utilisateurs doivent s'authentifier.

Voici un exemple abrégé :

```
<file>
<?xml version="1.0" encoding="ISO-8859-1"?>
<web-app xmlns="http://xmlns.jcp.org/xml/ns/javaee"
  xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
  xsi:schemaLocation="http://xmlns.jcp.org/xml/ns/javaee
                      http://xmlns.jcp.org/xml/ns/javaee/web-app_3_1.xsd"
  version="3.1"
  metadata-complete="true">

  <display-name>Tomcat Manager Application</display-name>
  <description>
    A scriptable management web application for the Tomcat Web Server;
    Manager lets you view, load/unload/etc particular web applications.
  </description>

  <servlet>
    <servlet-name>Manager</servlet-name>
    <servlet-class>org.apache.catalina.manager.ManagerServlet</servlet-class>
    <init-param>
      <param-name>debug</param-name>
      <param-value>2</param-value>
    </init-param>
  </servlet>
  <servlet>
    <servlet-name>HTMLManager</servlet-name>
    <servlet-class>org.apache.catalina.manager.HTMLManagerServlet</servlet-class>
    <init-param>
      <param-name>debug</param-name>
      <param-value>2</param-value>
    </init-param>
    <!-- Uncomment this to show proxy sessions from the Backup manager or a
         StoreManager in the sessions list for an application
    <init-param>
      <param-name>showProxySessions</param-name>
```

```
      <param-value>true</param-value>
    </init-param>
    -->
    <multipart-config>
      <!-- 50MB max -->
      <max-file-size>52428800</max-file-size>
      <max-request-size>52428800</max-request-size>
      <file-size-threshold>0</file-size-threshold>
    </multipart-config>
  </servlet>
  <servlet>
    <servlet-name>Status</servlet-name>
    <servlet-class>org.apache.catalina.manager.StatusManagerServlet</servlet-class>
    <init-param>
      <param-name>debug</param-name>
      <param-value>0</param-value>
    </init-param>
  </servlet>

  <servlet>
    <servlet-name>JMXProxy</servlet-name>
    <servlet-class>org.apache.catalina.manager.JMXProxyServlet</servlet-class>
  </servlet>

  <!-- Define the Manager Servlet Mapping -->
  <servlet-mapping>
    <servlet-name>Manager</servlet-name>
      <url-pattern>/text/*</url-pattern>
  </servlet-mapping>
  <servlet-mapping>
    <servlet-name>Status</servlet-name>
    <url-pattern>/status/*</url-pattern>
  </servlet-mapping>
  <servlet-mapping>
```

```
    <servlet-name>JMXProxy</servlet-name>
      <url-pattern>/jmxproxy/*</url-pattern>
  </servlet-mapping>
  <servlet-mapping>
    <servlet-name>HTMLManager</servlet-name>
    <url-pattern>/html/*</url-pattern>
  </servlet-mapping>

  <filter>
    <filter-name>SetCharacterEncoding</filter-name>
    <filter-class>org.apache.catalina.filters.SetCharacterEncodingFilter</filter-class>
    <init-param>
      <param-name>encoding</param-name>
      <param-value>UTF-8</param-value>
    </init-param>
  </filter>

  <filter-mapping>
    <filter-name>SetCharacterEncoding</filter-name>
    <url-pattern>/*</url-pattern>
  </filter-mapping>

  <filter>
    <filter-name>CSRF</filter-name>
    <filter-class>org.apache.catalina.filters.CsrfPreventionFilter</filter-class>
    <init-param>
      <param-name>entryPoints</param-name>
      <param-value>/html,/html/,/html/list,/index.jsp</param-value>
    </init-param>
  </filter>

  <filter-mapping>
    <filter-name>CSRF</filter-name>
    <servlet-name>HTMLManager</servlet-name>
```

```
  </filter-mapping>

  <!-- Define a Security Constraint on this Application -->
  <!-- NOTE:  None of these roles are present in the default users file -->
  <security-constraint>
    <web-resource-collection>
      <web-resource-name>HTML Manager interface (for humans)</web-resource-name>
      <url-pattern>/html/*</url-pattern>
    </web-resource-collection>
    <auth-constraint>
       <role-name>manager-gui</role-name>
    </auth-constraint>
  </security-constraint>
  <security-constraint>
    <web-resource-collection>
      <web-resource-name>Text Manager interface (for scripts)</web-resource-name>
      <url-pattern>/text/*</url-pattern>
    </web-resource-collection>
    <auth-constraint>
       <role-name>manager-script</role-name>
    </auth-constraint>
  </security-constraint>
  <security-constraint>
    <web-resource-collection>
      <web-resource-name>JMX Proxy interface</web-resource-name>
      <url-pattern>/jmxproxy/*</url-pattern>
    </web-resource-collection>
    <auth-constraint>
       <role-name>manager-jmx</role-name>
    </auth-constraint>
  </security-constraint>
  <security-constraint>
    <web-resource-collection>
      <web-resource-name>Status interface</web-resource-name>
```

```
    <url-pattern>/status/*</url-pattern>
  </web-resource-collection>
  <auth-constraint>
     <role-name>manager-gui</role-name>
     <role-name>manager-script</role-name>
     <role-name>manager-jmx</role-name>
     <role-name>manager-status</role-name>
  </auth-constraint>
</security-constraint>

<!-- Define the Login Configuration for this Application -->
<login-config>
  <auth-method>BASIC</auth-method>
  <realm-name>Tomcat Manager Application</realm-name>
</login-config>

<!-- Security roles referenced by this web application -->
<security-role>
  <description>
    The role that is required to access the HTML Manager pages
  </description>
  <role-name>manager-gui</role-name>
</security-role>
<security-role>
  <description>
    The role that is required to access the text Manager pages
  </description>
  <role-name>manager-script</role-name>
</security-role>
<security-role>
  <description>
    The role that is required to access the HTML JMX Proxy
  </description>
  <role-name>manager-jmx</role-name>
```

```
  </security-role>
  <security-role>
    <description>
      The role that is required to access to the Manager Status pages
    </description>
    <role-name>manager-status</role-name>
  </security-role>

  <error-page>
    <error-code>401</error-code>
    <location>/WEB-INF/jsp/401.jsp</location>
  </error-page>
  <error-page>
    <error-code>403</error-code>
    <location>/WEB-INF/jsp/403.jsp</location>
  </error-page>
  <error-page>
    <error-code>404</error-code>
    <location>/WEB-INF/jsp/404.jsp</location>
  </error-page>

</web-app>
```

On peut constater dans ce fichier la présence d'une en-tête qui spécifie la version **XML** :

```
<?xml version="1.0" encoding="ISO-8859-1"?>
```

Le fichier contient ensuite l'ouverture de la balise **<web-app…>** qui constitue l'élément racine du fichier. Il contient notamment le **schéma XML** utilisé pour valider le contenu du fichier. Un schéma XML est un fichier ayant une extension **.xsd**. L'élément **web-app** est fermé en fin de fichier avec la balise **</web-app>** :

```
<web-app xmlns="http://xmlns.jcp.org/xml/ns/javaee"
  xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
  xsi:schemaLocation="http://xmlns.jcp.org/xml/ns/javaee
```

```
                      http://xmlns.jcp.org/xml/ns/javaee/web-app_3_1.xsd"
  version="3.1"
  metadata-complete="true">
...
</web-app>
```

L'élément **<display-name>** donne ensuite un nom à l'application :

```
  <display-name>Tomcat Manager Application</display-name>
```

L'élément **<description>** indique une description de l'application :

```
  <description>
    A scriptable management web application for the Tomcat Web Server;
    Manager lets you view, load/unload/etc particular web applications.
  </description>
```

L'élément **<filter>** permet de spécifier le code de caractères à utiliser pour l'application :

```
  <filter>
    <filter-name>SetCharacterEncoding</filter-name>
    <filter-class>org.apache.catalina.filters.SetCharacterEncodingFilter</filter-class>
    <init-param>
      <param-name>encoding</param-name>
      <param-value>UTF-8</param-value>
    </init-param>
  </filter>
```

Ensuite l'élément **<filter>** déclaré doit être associé à une URL grâce à l'élément **<filter-mapping>** :

```
  <filter-mapping>
    <filter-name>SetCharacterEncoding</filter-name>
    <url-pattern>/*</url-pattern>
```

```
  </filter-mapping>
```

Chaque servlet doit ensuite être déclarée en donnant la correspondance entre un **nom logique** et la classe Java de la servlet. La déclaration peut également contenir des paramètres d'initialisation propre à la servlet. Si les paramètres d'initialisation sont déclarées en dehors d'un élément **<servlet>** dans un élément **<context-param>**, ces paramètres s'appliquent à toutes les servlets et à toutes les JSP de l'application :

```
    <servlet>
    <servlet-name>Manager</servlet-name>
    <servlet-class>org.apache.catalina.manager.ManagerServlet</servlet-class>
    <init-param>
      <param-name>debug</param-name>
      <param-value>2</param-value>
    </init-param>
  </servlet>
  <servlet>
    <servlet-name>HTMLManager</servlet-name>
    <servlet-class>org.apache.catalina.manager.HTMLManagerServlet</servlet-class>
    <init-param>
      <param-name>debug</param-name>
      <param-value>2</param-value>
    </init-param>
    <multipart-config>
      <max-file-size>52428800</max-file-size>
      <max-request-size>52428800</max-request-size>
      <file-size-threshold>0</file-size-threshold>
    </multipart-config>
  </servlet>
  <servlet>
    <servlet-name>Status</servlet-name>
    <servlet-class>org.apache.catalina.manager.StatusManagerServlet</servlet-class>
    <init-param>
      <param-name>debug</param-name>
      <param-value>0</param-value>
    </init-param>
```

```
  </servlet>

  <servlet>
    <servlet-name>JMXProxy</servlet-name>
    <servlet-class>org.apache.catalina.manager.JMXProxyServlet</servlet-class>
  </servlet>
```

Ensuite chaque servlet déclarée doit être associée à une URL grâce à l'élément **<servlet-mapping>** :

```
  <servlet-mapping>
    <servlet-name>Manager</servlet-name>
      <url-pattern>/text/*</url-pattern>
  </servlet-mapping>
  <servlet-mapping>
    <servlet-name>Status</servlet-name>
    <url-pattern>/status/*</url-pattern>
  </servlet-mapping>
  <servlet-mapping>
    <servlet-name>JMXProxy</servlet-name>
      <url-pattern>/jmxproxy/*</url-pattern>
  </servlet-mapping>
  <servlet-mapping>
    <servlet-name>HTMLManager</servlet-name>
    <url-pattern>/html/*</url-pattern>
  </servlet-mapping>
```

Finalement l'élément **<security-constraint>** est utilisé pour restreindre l'accès à certaines parties de l'application à certains utilisateurs :

```
<security-constraint>
   <web-resource-collection>
     <web-resource-name>HTML Manager interface (for humans)</web-resource-name>
     <url-pattern>/html/*</url-pattern>
   </web-resource-collection>
   <auth-constraint>
```

```
      <role-name>manager-gui</role-name>
    </auth-constraint>
  </security-constraint>
  <security-constraint>
    <web-resource-collection>
      <web-resource-name>Text Manager interface (for scripts)</web-resource-name>
      <url-pattern>/text/*</url-pattern>
    </web-resource-collection>
    <auth-constraint>
       <role-name>manager-script</role-name>
    </auth-constraint>
  </security-constraint>
  <security-constraint>
    <web-resource-collection>
      <web-resource-name>JMX Proxy interface</web-resource-name>
      <url-pattern>/jmxproxy/*</url-pattern>
    </web-resource-collection>
    <auth-constraint>
       <role-name>manager-jmx</role-name>
    </auth-constraint>
  </security-constraint>
  <security-constraint>
    <web-resource-collection>
      <web-resource-name>Status interface</web-resource-name>
      <url-pattern>/status/*</url-pattern>
    </web-resource-collection>
    <auth-constraint>
       <role-name>manager-gui</role-name>
       <role-name>manager-script</role-name>
       <role-name>manager-jmx</role-name>
       <role-name>manager-status</role-name>
    </auth-constraint>
  </security-constraint>
  <login-config>
```

```
    <auth-method>BASIC</auth-method>
    <realm-name>Tomcat Manager Application</realm-name>
  </login-config>
  <security-role>
    <description>
      The role that is required to access the HTML Manager pages
    </description>
    <role-name>manager-gui</role-name>
  </security-role>
  <security-role>
    <description>
      The role that is required to access the text Manager pages
    </description>
    <role-name>manager-script</role-name>
  </security-role>
  <security-role>
    <description>
      The role that is required to access the HTML JMX Proxy
    </description>
    <role-name>manager-jmx</role-name>
  </security-role>
  <security-role>
    <description>
      The role that is required to access to the Manager Status pages
    </description>
    <role-name>manager-status</role-name>
  </security-role>
```

Les Sessions HTTP

Les **Sessions HTTP** permettent au serveur d'identifier un client lors de sa navigation. Chaque session est associée à un identifiant unique qui généré par le serveur et est envoyé vers le client. Quand le client envoie une requête, il inclut l'identifiant de la session.

L'identifiant de session peut être envoyé au client soit par le biais d'un cookie nommé **jsessionid**, soit inclus dans l'URLs envoyées par le serveur vers le client dans le cas où ce dernier n'accepte pas les cookies.

Les cookies :

- sont des informations transmises sous format texte,
- sont limités à 20 par site et 300 par navigateur,
- sont limités en taille à 4Ko.

La session expire à la fermeture du navigateur web ou après 30 minutes d'inactivité.

SER302 - Installation de Tomcat 8 et les serveurs associés

Contenu du Module

Désactiver SELinux

Afin d'éviter à ce que SELinux interfère avec l'installation et la configuration de Tomcat 8, éditez le fichier **/etc/selinux/config** afin de le désactiver :

```
[root@centos7 ~]# vi /etc/selinux/config
[root@centos7 ~]# cat /etc/selinux/config

# This file controls the state of SELinux on the system.
# SELINUX= can take one of these three values:
#     enforcing - SELinux security policy is enforced.
#     permissive - SELinux prints warnings instead of enforcing.
#     disabled - No SELinux policy is loaded.
SELINUX=disabled
# SELINUXTYPE= can take one of three two values:
#     targeted - Targeted processes are protected,
#     minimum - Modification of targeted policy. Only selected processes are protected.
#     mls - Multi Level Security protection.
SELINUXTYPE=targeted
```

Redémarrez ensuite le système :

```
[root@centos7 ~]# shutdown -r now
```

Tomcat et JDK

Pour installer Tomcat 8 sous **CentOS 7**, il convient de saisir les commandes suivantes :

```
[root@centos7 ~]# wget https://archive.apache.org/dist/tomcat/tomcat-8/v8.0.36/bin/apache-tomcat-8.0.36.tar.gz
```

Puis :

```
[root@centos7 ~]# tar xvf apache-tomcat-8.0.36.tar.gz
[root@centos7 ~]# mv apache-tomcat-8.0.36 /usr/tomcat8
```

Installez maintenant le JDK :

```
[root@centos7 ~]# yum install java-1.8.0-openjdk-devel
```

Vérifiez la présence du **jre-1.8.0-openjdk** dans le répertoire **/usr/lib/jvm** :

```
[root@centos7 ~]# ls -l /usr/lib/jvm/jre-1.8.0-openjdk*
lrwxrwxrwx. 1 root root 35 Oct 28 18:23 /usr/lib/jvm/jre-1.8.0-openjdk -> /etc/alternatives/jre_1.8.0_openjdk
lrwxrwxrwx. 1 root root 51 Oct 28 18:03 /usr/lib/jvm/jre-1.8.0-openjdk-1.8.0.232.b09-0.el7_7.x86_64 ->
java-1.8.0-openjdk-1.8.0.232.b09-0.el7_7.x86_64/jre
```

Ajoutez les trois lignes suivantes au fichier **/etc/profile** :

```
...
PATH=$PATH:/usr/lib/jvm/jre-1.8.0-openjdk-1.8.0.232.b09-0.el7_7.x86_64/bin
JAVA_HOME=/usr/lib/jvm/jre-1.8.0-openjdk-1.8.0.232.b09-0.el7_7.x86_64
export PATH JAVA_HOME
```

Important : Vérifiez que la version du **jre-1.8.0-openjdk** dans le fichier **/etc/profile** est la même que dans le répertoire **/usr/lib/jvm**.

Rechargez le fichier /etc/profile et vérifiez les valeurs des deux variables précédemment déclarées :

```
[root@centos7 ~]# source /etc/profile
[root@centos7 ~]# echo $PATH
/usr/local/sbin:/usr/local/bin:/sbin:/bin:/usr/sbin:/usr/bin:/root/bin:/usr/lib/jvm/jre-1.8.0-
openjdk-1.8.0.232.b09-0.el7_7.x86_64/bin
[root@centos7 ~]# echo $JAVA_HOME
```

```
/usr/lib/jvm/jre-1.8.0-openjdk-1.8.0.232.b09-0.el7_7.x86_64
```

Vérifiez ensuite la version de java :

```
[root@centos7 ~]# java -version
openjdk version "1.8.0_232"
OpenJDK Runtime Environment (build 1.8.0_232-b09)
OpenJDK 64-Bit Server VM (build 25.232-b09, mixed mode)
```

Définissez maintenant la variable CATALINA_HOME dans le fichier **/etc/profile** :

```
...
# Tomcat
CATALINA_HOME="/usr/tomcat8"
export CATALINA_HOME
PATH=$PATH:/usr/lib/jvm/jre-1.8.0-openjdk-1.8.0.232.b09-0.el7_7.x86_64/bin
JAVA_HOME=/usr/lib/jvm/jre-1.8.0-openjdk-1.8.0.232.b09-0.el7_7.x86_64
export PATH JAVA_HOME
```

Rechargez /etc/profile :

```
[root@centos7 ~]# source /etc/profile
[root@centos7 ~]# echo $CATALINA_HOME
/usr/tomcat8
```

Démarrez maintenant Tomcat 8 :

```
[root@centos7 ~]# cd /usr/tomcat8/bin
[root@centos7 bin]# ./startup.sh
Using CATALINA_BASE:   /usr/tomcat8
Using CATALINA_HOME:   /usr/tomcat8
Using CATALINA_TMPDIR: /usr/tomcat8/temp
Using JRE_HOME:        /usr/lib/jvm/jre-1.8.0-openjdk-1.8.0.232.b09-0.el7_7.x86_64
Using CLASSPATH:       /usr/tomcat8/bin/bootstrap.jar:/usr/tomcat8/bin/tomcat-juli.jar
```

```
Tomcat started.
```

Utilisez le navigateur de texte **lynx** pour tester Tomcat 8 :

```
[root@centos7 bin]# yum install lynx
...
[root@centos7 bin]# lynx --dump http://localhost:8080
   [1]Home [2]Documentation [3]Configuration [4]Examples [5]Wiki
   [6]Mailing Lists [7]Find Help

Apache Tomcat/8.0.36

If you're seeing this, you've successfully installed Tomcat. Congratulations!

   [tomcat logo]

Recommended Reading:
...
```

Apache

Présentation d'Apache

Un serveur web est une machine doté d'un logiciel serveur qui attend des requêtes de la part de machines clientes afin de leur livrer de documents de types différents.

En 1994 le développement du serveur web le plus connue à l'époque, le démon **HTTP**, a été arrêté suite au départ de la NCSA de son principal développeur, **Rob McCool**.

Au début de l'année 1995, un groupe de webmestres indépendants s'est mis en place sous la direction de **Brian Behlendorf** et **Cliff Skolnick** pour reprendre le travail sur ce démon. Ce projet a pris le nom **Apache**. En même temps la NCSA a repris son propre travail de développement sur son démon HTTP. L'arrivée dans le groupe Apache de deux personnes de la NCSA en tant que membres honoraires, **Brandon Long** et **Beth Frank** a

permis la mise en commun des connaissances des deux groupes.

Le projet **Apache** est un projet de développement d'un serveur web libre pour les plateformes Unix et Windows™. La première version *officielle*, la 0.6.2 est sortie en avril 1995.

La **Fondation Apache**, créée en 1999 par l'équipe Apache, gère aujourd'hui non seulement le projet Apache mais aussi un grand nombre d'autres projets. La liste des projets de la Fondation peut être trouvée **ici**.

Apache est modulaire. Certains modules fondamentaux conditionnent comment Apache traite la question du multitraitement. Les modules multitraitements - **MPM - Multi-Processing Modules** - sont différents selon le système d'exploitation utilisé et la charge attendue.

- **mpm-winnt** - module propre à Windows™ qui utilise son support réseau natif,
- **prefork** - module propre à Unix et Linux qui implémente un serveur mono-tâche à duplication,
- **perchild** - module propre à Unix et Linux qui implémente un serveur autorisant des démons servant les requêtes à être assigner à plusieurs id utilisateurs,
- **worker** - module propre à Unix et Linux qui implémente un serveur hybride multi-tâche et multitraitement.

Ces modules sont compilés statiquement au binaire Apache et sont mutuellement exclusifs.

Installation

Sous **CentOS 7**, Apache n'est pas installé par défaut :

```
[root@centos7 bin]# cd ~
[root@centos7 ~]# rpm -qa | grep httpd
[root@centos7 ~]#
[root@centos7 ~]# yum install httpd
```

Le service n'est pas configuré pour démarrer automatiquement :

```
[root@centos7 ~]# systemctl status httpd
● httpd.service - The Apache HTTP Server
   Loaded: loaded (/usr/lib/systemd/system/httpd.service; disabled; vendor preset: disabled)
```

```
   Active: inactive (dead)
     Docs: man:httpd(8)
           man:apachectl(8)
```

Saisissez donc les commandes suivantes et vérifiez le résultat :

```
[root@centos7 ~]# systemctl enable httpd.service
Created symlink from /etc/systemd/system/multi-user.target.wants/httpd.service to
/usr/lib/systemd/system/httpd.service.
[root@centos7 ~]# systemctl start httpd.service
[root@centos7 ~]# systemctl status httpd.service
● httpd.service - The Apache HTTP Server
   Loaded: loaded (/usr/lib/systemd/system/httpd.service; enabled; vendor preset: disabled)
   Active: active (running) since Mon 2019-10-28 19:02:22 CET; 5s ago
     Docs: man:httpd(8)
           man:apachectl(8)
 Main PID: 7427 (httpd)
   Status: "Processing requests..."
   CGroup: /system.slice/httpd.service
           ├─7427 /usr/sbin/httpd -DFOREGROUND
           ├─7455 /usr/sbin/httpd -DFOREGROUND
           ├─7456 /usr/sbin/httpd -DFOREGROUND
           ├─7457 /usr/sbin/httpd -DFOREGROUND
           ├─7458 /usr/sbin/httpd -DFOREGROUND
           └─7459 /usr/sbin/httpd -DFOREGROUND

Oct 28 19:02:19 centos7.fenestros.loc systemd[1]: Starting The Apache HTTP Server...
Oct 28 19:02:22 centos7.fenestros.loc systemd[1]: Started The Apache HTTP Server.
```

Testez le serveur apache avec telnet

Telnet n'est pas installé par défaut sous CentOS 7 :

```
[root@centos7 ~]# which telnet
/usr/bin/which: no telnet in
(/usr/local/sbin:/usr/local/bin:/sbin:/bin:/usr/sbin:/usr/bin:/root/bin:/usr/lib/jvm/jre-1.8.0-
openjdk-1.8.0.232.b09-0.el7_7.x86_64/bin:/usr/lib/jvm/jre-1.8.0-openjdk-1.8.0.232.b09-0.el7_7.x86_64/bin)
```

Installez donc telnet :

```
[root@centos7 ~]# yum install telnet
```

Testez maintenant le service Apache :

```
[root@centos7 ~]# telnet localhost 80
Trying 127.0.0.1...
Connected to localhost.
Escape character is '^]'.
```

Tapez ensuite **GET /** :

```
GET /
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.1//EN" "http://www.w3.org/TR/xhtml11/DTD/xhtml11.dtd"><html><head>
<meta http-equiv="content-type" content="text/html; charset=UTF-8">
        <title>Apache HTTP Server Test Page powered by CentOS</title>
        <meta http-equiv="Content-Type" content="text/html; charset=UTF-8">

    <!-- Bootstrap -->
    <link href="/noindex/css/bootstrap.min.css" rel="stylesheet">
    <link rel="stylesheet" href="noindex/css/open-sans.css" type="text/css" />

<style type="text/css"><!--

body {
  font-family: "Open Sans", Helvetica, sans-serif;
  font-weight: 100;
```

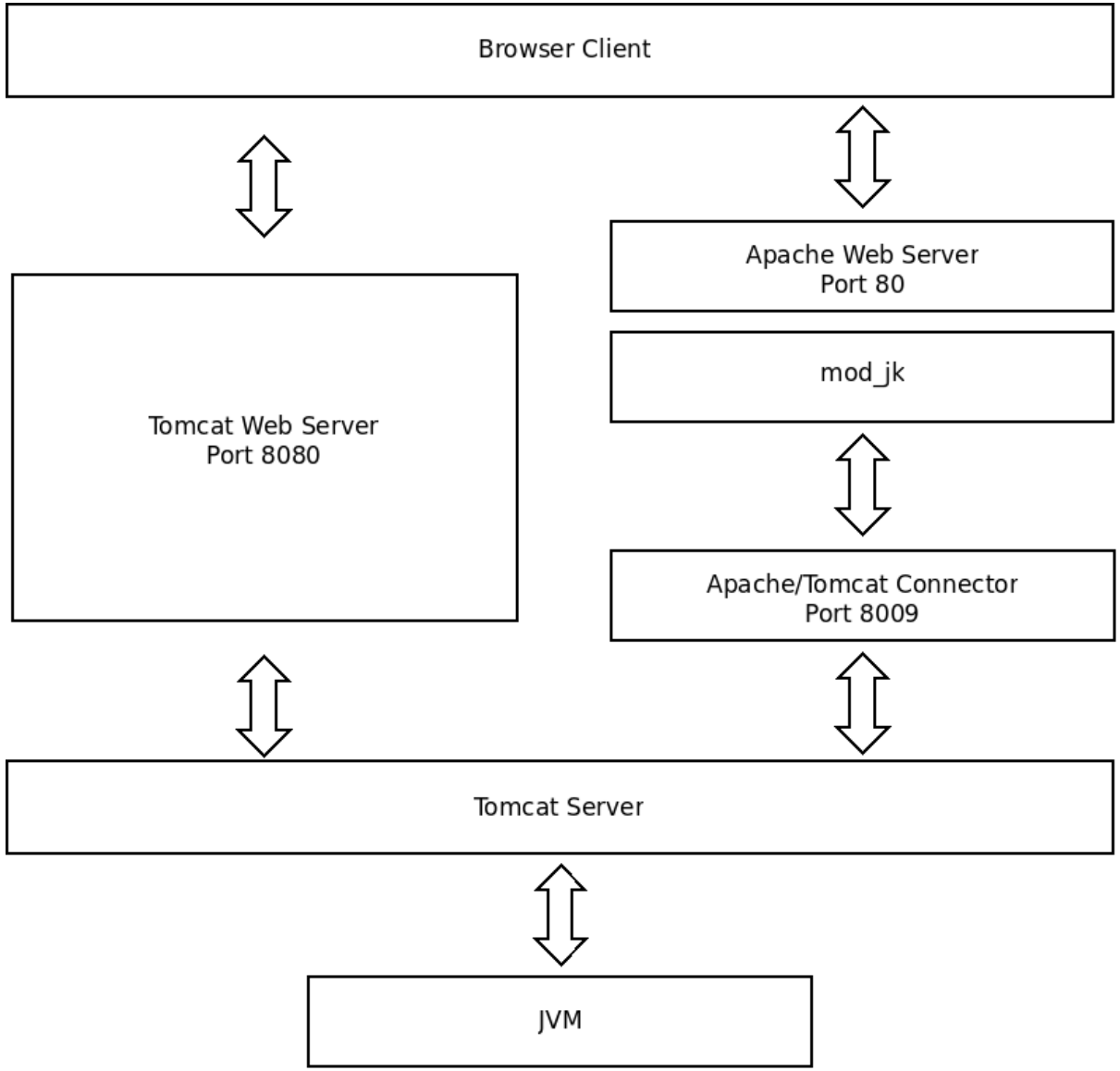
```
  color: #ccc;
  background: rgba(10, 24, 55, 1);
  font-size: 16px;
}

h2, h3, h4 {
...
```

Pour voir la version d'Apache installée, utilisez la commande suivante :

```
[root@centos7 ~]# httpd -v
Server version: Apache/2.4.6 (CentOS)
Server built:   Aug  8 2019 11:41:18
```

Coupler Tomcat et Apache

Le schéma suivant indique le couplage Tomcat/Apache :



Téléchargez le connecteur mod-jk pour Apache :

```
[root@centos7 ~]# wget
```

```
http://apache.mirrors.ovh.net/ftp.apache.org/dist/tomcat/tomcat-connectors/jk/tomcat-connectors-1.2.48-src.tar.gz
```

Désarchivez l'archive et placez-vous dans le répertoire **tomcat-connectors-1.2.48-src** :

```
[root@centos7 ~]# tar xvf tomcat-connectors-1.2.48-src.tar.gz
[root@centos7 ~]# cd tomcat-connectors-1.2.48-src/
```

Installez le binaire **apxs** inclut dans le paquet **httpd-devel**. Le binaire apxs est utilisé pour construire les modules d'Apache :

```
[root@centos7 tomcat-connectors-1.2.48-src]# yum install httpd-devel
```

Placez-vous dans le sous-répertoire **native** et lancez les commandes pour effectuer la compilation :

```
[root@centos7 tomcat-connectors-1.2.48-src]# cd native/
[root@centos7 native]# which apxs
/bin/apxs
[root@centos7 native]# ./configure --with-apxs=/bin/apxs
...
[root@centos7 native]# make
...
[root@centos7 native]# make install
Making install in common
make[1]: Entering directory `/root/tomcat-connectors-1.2.46-src/native/common'
make[1]: Nothing to be done for `install'.
make[1]: Leaving directory `/root/tomcat-connectors-1.2.46-src/native/common'
Making install in apache-2.0
make[1]: Entering directory `/root/tomcat-connectors-1.2.46-src/native/apache-2.0'

Installing files to Apache Modules Directory...
/bin/apxs -i mod_jk.la
/usr/lib64/httpd/build/instdso.sh SH_LIBTOOL='/usr/lib64/apr-1/build/libtool' mod_jk.la /usr/lib64/httpd/modules
/usr/lib64/apr-1/build/libtool --mode=install install mod_jk.la /usr/lib64/httpd/modules/
libtool: install: install .libs/mod_jk.so /usr/lib64/httpd/modules/mod_jk.so
libtool: install: install .libs/mod_jk.lai /usr/lib64/httpd/modules/mod_jk.la
```

```
libtool: install: install .libs/mod_jk.a /usr/lib64/httpd/modules/mod_jk.a
libtool: install: chmod 644 /usr/lib64/httpd/modules/mod_jk.a
libtool: install: ranlib /usr/lib64/httpd/modules/mod_jk.a
libtool: finish:
PATH="/usr/local/sbin:/usr/local/bin:/sbin:/bin:/usr/sbin:/usr/bin:/root/bin:/usr/lib/jvm/jre-1.8.0-
openjdk-1.8.0.232.b09-0.el7_7.x86_64/bin:/usr/lib/jvm/jre-1.8.0-
openjdk-1.8.0.232.b09-0.el7_7.x86_64/bin:/usr/lib/jvm/jre-1.8.0-
openjdk-1.8.0.232.b09-0.el7_7.x86_64/bin:/usr/lib/jvm/jre-1.8.0-openjdk-1.8.0.232.b09-0.el7_7.x86_64/bin:/sbin"
ldconfig -n /usr/lib64/httpd/modules
----------------------------------------------------------------------
Libraries have been installed in:
   /usr/lib64/httpd/modules

If you ever happen to want to link against installed libraries
in a given directory, LIBDIR, you must either use libtool, and
specify the full pathname of the library, or use the `-LLIBDIR'
flag during linking and do at least one of the following:
   - add LIBDIR to the `LD_LIBRARY_PATH' environment variable
     during execution
   - add LIBDIR to the `LD_RUN_PATH' environment variable
     during linking
   - use the `-Wl,-rpath -Wl,LIBDIR' linker flag
   - have your system administrator add LIBDIR to `/etc/ld.so.conf'

See any operating system documentation about shared libraries for
more information, such as the ld(1) and ld.so(8) manual pages.
----------------------------------------------------------------------
chmod 755 /usr/lib64/httpd/modules/mod_jk.so

Please be sure to arrange /etc/httpd/conf/httpd.conf...

make[1]: Leaving directory `/root/tomcat-connectors-1.2.46-src/native/apache-2.0'
make[1]: Entering directory `/root/tomcat-connectors-1.2.46-src/native'
make[2]: Entering directory `/root/tomcat-connectors-1.2.46-src/native'
```

```
make[2]: Nothing to be done for `install-exec-am'.
make[2]: Nothing to be done for `install-data-am'.
make[2]: Leaving directory `/root/tomcat-connectors-1.2.46-src/native'
make[1]: Leaving directory `/root/tomcat-connectors-1.2.46-src/native'
```

Modifiez ensuite le fichier /etc/httpd/conf/httpd.conf ainsi :

```
...
ServerName www.i2tch.loc:80
...
```

et ajoutez les directives suivantes à la fin du fichier afin de prendre en compte mod_jk et sa configuration :

```
...
LoadModule jk_module modules/mod_jk.so
JkWorkersFile   conf/workers.properties
JkLogFile   logs/mod_jk.log
JkLogLevel  info
JkMount     /docs/*    worker1
JkMount     /docs    worker1
```

Les différentes directives Apache utilisables avec **mod_jk** sont :

Directive	**Description**
JkWorkersFile	Spécifie le nom et l'emplacement du fichier de configuration du module nommé **workers.properties** par convention. Le chemin peut être absolu ou rélatif à l'installation d'Apache.
JkWorkerProperty	L'utilisation de cette directive est mutuellement exclusive avec l'utilisation de la directive **JkWorkersFile**. Elle permet de définir les directives du fichier **workers.properties** directement dans le fichier **httpd.conf** sous la forme **JkWorkerProperty <Directive>** avec une directive par ligne. Chaque ligne doit donc commencer par **JkWorkerProperty**.
JkLogFile	Spécifie le nom et l'emplacement du fichier de journalisation du module. Le chemin peut être absolu ou relatif à l'installation d'Apache.
JkLogLevel	Spécifie le niveau de journalisation du module. Les valeurs ici peuvent être **trace**, **debug**, **info**, **warn** et **error**. Le fonctionnement est similaire à syslog dans la mesure où si la valeur est **info** alors tous les messages des niveaux **info**, **warn** et **error** seront journalisés.

Directive	Description
JkMount	Spécifie l'association d'un contexte d'application Tomcat avec un travailleur. Autrement dit, cette directive spécifie quelles sont les ressources d'une application Tomcat accessibles en passant par Apache. La syntaxe est **JkMount <URL> <Nom du Travailleur>**.
JkUnMount	Spécifie une interdiction de redirection de requêtes vers une ressource d'une application Tomcat. La syntaxe est **JkUnMount <URL> <Nom du Travailleur>**. Notez que cette directive est prioritaire par rapport à la directive **JkMount**.
JkAutoAlias	Spécifie un alias entre le répertoire des applications de Tomcat et le répertoire de publication des fichiers statiques d'Apache. De cette façon, les deux serveurs partagent un répertoire de publication unique.
JkLogStampFormat	Spécifie le format de la date inscrite dans le fichier de journalisation du module en utilisant des séquences de contrôle, par exemple, **JkLogStampFormat "[%a %b %d %H:%M:%S %Y]"**.
JkExtractSSL	Spécifie que le module peut transmettre les informations SSL vers le serveur Tomcat.
JkHTTPSIndicator	Spécifie le nom de la variable Apache contenant l'indication SSL.
JkCERTSIndicator	Spécifie le nom de la variable Apache contenant le certificat SSL.
JkSESSIONIndicator	Spécifie le nom de la variable Apache contenant l'identifiant de session SSL.

Modifiez ensuite le fichier **/etc/hosts** pour la prise en charge du nom de notre serveur :

```
[root@centos7 native]# vi /etc/hosts
[root@centos7 native]# cat /etc/hosts
127.0.0.1       localhost.localdomain localhost
::1     localhost6.localdomain6 localhost6
10.0.2.51   www.i2tch.loc
```

Créez ensuite le fichier **/etc/httpd/conf/workers.properties** :

```
[root@centos7 native]# cd ~
[root@centos7 ~]# vi /etc/httpd/conf/workers.properties
[root@centos7 ~]# cat /etc/httpd/conf/workers.properties
worker.list=worker1
worker.worker1.type=ajp13
worker.worker1.host=10.0.2.51
worker.worker1.port=8009
```

Vérifiez l'existence des lignes concernant le connecteur dans le fichier **/usr/tomcat8/conf/server.xml** :

```
[root@centos7 ~]# cat /usr/tomcat8/conf/server.xml | grep 8009
    <!-- Define an AJP 1.3 Connector on port 8009 -->
    <Connector port="8009" protocol="AJP/1.3" redirectPort="8443" />
```

Redémarrez ensuite les services httpd et tomcat8 :

```
[root@centos7 ~]# systemctl restart httpd.service
[root@centos7 ~]# cd /usr/tomcat8/bin/
[root@centos7 bin]# ls
bootstrap.jar  catalina-tasks.xml           configtest.bat  digest.bat        setclasspath.sh  startup.bat
tomcat-native.tar.gz  version.bat
catalina.bat   commons-daemon.jar           configtest.sh   digest.sh         shutdown.bat     startup.sh
tool-wrapper.bat      version.sh
catalina.sh    commons-daemon-native.tar.gz  daemon.sh       setclasspath.bat  shutdown.sh      tomcat-juli.jar
tool-wrapper.sh
[root@centos7 ~]# cd /usr/tomcat8/bin/
[root@centos7 bin]# ./shutdown.sh
Using CATALINA_BASE:   /usr/tomcat8
Using CATALINA_HOME:   /usr/tomcat8
Using CATALINA_TMPDIR: /usr/tomcat8/temp
Using JRE_HOME:        /usr/lib/jvm/jre-1.8.0-openjdk-1.8.0.232.b09-0.el7_7.x86_64
Using CLASSPATH:       /usr/tomcat8/bin/bootstrap.jar:/usr/tomcat8/bin/tomcat-juli.jar
[root@centos7 bin]# ./startup.sh
Using CATALINA_BASE:   /usr/tomcat8
Using CATALINA_HOME:   /usr/tomcat8
Using CATALINA_TMPDIR: /usr/tomcat8/temp
Using JRE_HOME:        /usr/lib/jvm/jre-1.8.0-openjdk-1.8.0.232.b09-0.el7_7.x86_64
Using CLASSPATH:       /usr/tomcat8/bin/bootstrap.jar:/usr/tomcat8/bin/tomcat-juli.jar
Tomcat started.
```

Vérifiez maintenant le bon fonctionnement d'Apache et de Tomcat en consultant les liens suivants :

```
[root@centos7 bin]# lynx --dump http://www.i2tch.loc
[root@centos7 bin]# lynx --dump http://www.i2tch.loc:8080
[root@centos7 bin]# lynx --dump http://www.i2tch.loc/docs/
```

> **Important** : Notez qu'en consultant l'adresse **http://www.i2tch.loc/docs/** vous obtenez la même page que lors de votre consultation de l'adresse **http://www.i2tch.loc:8080**. Ceci implique que le serveur Apache est maintenant un **serveur frontal** pour le serveur Tomcat 8.

Créer un Service Systemd pour Tomcat

Créez un utilisateur tomcat :

```
[root@centos7 ~]# useradd -M -s /bin/nologin -d /usr/tomcat8 tomcat
```

Naviguez à votre répertoire $CATALINA_HOME :

```
[root@centos7 ~]# cd $CATALINA_HOME
[root@centos7 tomcat8]#
```

Modifiez les droits sur certains fichiers et répertoires :

```
[root@centos7 tomcat8]# chgrp -R tomcat conf
[root@centos7 tomcat8]# chmod g+rwx conf
[root@centos7 tomcat8]# chmod g+r conf/*
[root@centos7 tomcat8]# chown -R tomcat logs/ temp/ webapps/ work/
[root@centos7 tomcat8]# chgrp -R tomcat bin
[root@centos7 tomcat8]# chgrp -R tomcat lib
[root@centos7 tomcat8]# chmod g+rwx bin
[root@centos7 tomcat8]# chmod g+r bin/*
```

Créez le fichier de service **/etc/systemd/system/tomcat.service** :

```
[root@centos7 tomcat8]# vi /etc/systemd/system/tomcat.service
[root@centos7 tomcat8]# cat /etc/systemd/system/tomcat.service
# Systemd unit file for tomcat
[Unit]
Description=Apache Tomcat Web Application Container
After=syslog.target network.target

[Service]
Type=forking

Environment=JAVA_HOME="/usr/lib/jvm/jre-1.8.0-openjdk-1.8.0.232.b09-0.el7_7.x86_64"
Environment=CATALINA_PID=/usr/tomcat8/temp/tomcat.pid
Environment=CATALINA_HOME=/usr/tomcat8/
Environment=CATALINA_BASE=/usr/tomcat8/
Environment='CATALINA_OPTS=-Xms512M -Xmx1024M -server -XX:+UseParallelGC'
Environment='JAVA_OPTS=-Djava.awt.headless=true -Djava.security.egd=file:/dev/./urandom'

ExecStart=/usr/tomcat8/bin/startup.sh
ExecStop=/bin/kill -15 $MAINPID

User=tomcat
Group=tomcat

[Install]
WantedBy=multi-user.target
```

Redémarrez systemd :

```
[root@centos7 tomcat8]# systemctl daemon-reload
```

Arrêtez Tomcat :

```
[root@centos7 tomcat8]# cd bin
[root@centos7 bin]# ./shutdown.sh
Using CATALINA_BASE:   /usr/tomcat8
Using CATALINA_HOME:   /usr/tomcat8
Using CATALINA_TMPDIR: /usr/tomcat8/temp
Using JRE_HOME:        /usr/lib/jvm/jre-1.8.0-openjdk-1.8.0.232.b09-0.el7_7.x86_64
Using CLASSPATH:       /usr/tomcat8/bin/bootstrap.jar:/usr/tomcat8/bin/tomcat-juli.jar
```

Testez le service tomcat :

```
[root@centos7 tomcat8]# systemctl start tomcat
```

Vérifiez que le Tomcat a démarré :

```
[root@centos7 bin]# systemctl start tomcat
[root@centos7 bin]# ps aux | grep tomcat
tomcat   20279 27.3 25.9 3118244 129632 ?      Sl   10:58   0:02 /usr/lib/jvm/jre-1.8.0-
openjdk-1.8.0.232.b09-0.el7_7.x86_64/bin/java -
Djava.util.logging.config.file=/usr/tomcat8//conf/logging.properties -
Djava.util.logging.manager=org.apache.juli.ClassLoaderLogManager -Djava.awt.headless=true -
Djava.security.egd=file:/dev/./urandom -Djdk.tls.ephemeralDHKeySize=2048 -Xms512M -Xmx1024M -server -
XX:+UseParallelGC -Djava.endorsed.dirs=/usr/tomcat8//endorsed -classpath
/usr/tomcat8//bin/bootstrap.jar:/usr/tomcat8//bin/tomcat-juli.jar -Dcatalina.base=/usr/tomcat8/ -
Dcatalina.home=/usr/tomcat8/ -Djava.io.tmpdir=/usr/tomcat8//temp org.apache.catalina.startup.Bootstrap start
root     20376  0.0  0.1 112712   968 pts/0    R+   10:58   0:00 grep --color=auto tomcat
```

Dernièrement vérifiez que les serveur Tomcat est bien démarré sous la responsabilité de l'utilisateur tomcat :

```
[root@centos7 bin]# pgrep -u tomcat
20279
```

MariaDB

Présentation

MariaDB est un fork de **MySQL** et est inclut dans les dépôts de **CentOS 7**.

Installation

Pour installer MariaDB, utilisez yum :

```
[root@centos7 bin]# cd ~
[root@centos7 ~]# yum install mariadb mariadb-server
```

Ensuite démarrez votre serveur MariaDB :

```
[root@centos7 ~]# systemctl status mariadb.service
● mariadb.service - MariaDB database server
   Loaded: loaded (/usr/lib/systemd/system/mariadb.service; disabled; vendor preset: disabled)
   Active: inactive (dead)
[root@centos7 ~]# systemctl enable mariadb.service
Created symlink from /etc/systemd/system/multi-user.target.wants/mariadb.service to
/usr/lib/systemd/system/mariadb.service.
[root@centos7 ~]# systemctl start mariadb.service
[root@centos7 ~]# systemctl status mariadb.service
● mariadb.service - MariaDB database server
   Loaded: loaded (/usr/lib/systemd/system/mariadb.service; enabled; vendor preset: disabled)
   Active: active (running) since Thu 2017-04-27 13:44:02 CEST; 3s ago
  Process: 30380 ExecStartPost=/usr/libexec/mariadb-wait-ready $MAINPID (code=exited, status=0/SUCCESS)
  Process: 30295 ExecStartPre=/usr/libexec/mariadb-prepare-db-dir %n (code=exited, status=0/SUCCESS)
 Main PID: 30379 (mysqld_safe)
   CGroup: /system.slice/mariadb.service
```

```
           ├─30379 /bin/sh /usr/bin/mysqld_safe --basedir=/usr
           └─30536 /usr/libexec/mysqld --basedir=/usr --datadir=/var/lib/mysql --plugin-
dir=/usr/lib64/mysql/plugin --log-error=/var/log/mariadb/mariadb.l...

Apr 27 13:44:00 centos7.fenestros.loc mariadb-prepare-db-dir[30295]: OK
Apr 27 13:44:00 centos7.fenestros.loc mariadb-prepare-db-dir[30295]: To start mysqld at boot time you have to
copy
Apr 27 13:44:00 centos7.fenestros.loc mariadb-prepare-db-dir[30295]: support-files/mysql.server to the right
place for your system
Apr 27 13:44:00 centos7.fenestros.loc mariadb-prepare-db-dir[30295]: PLEASE REMEMBER TO SET A PASSWORD FOR THE
MariaDB root USER !
Apr 27 13:44:00 centos7.fenestros.loc mariadb-prepare-db-dir[30295]: To do so, start the server, then issue the
following commands:
Apr 27 13:44:00 centos7.fenestros.loc mariadb-prepare-db-dir[30295]: '/usr/bin/mysqladmin' -u root password 'new-
password'
Apr 27 13:44:00 centos7.fenestros.loc mariadb-prepare-db-dir[30295]: '/usr/bin/mysqladmin' -u root -h
centos7.fenestros.loc password 'new-password'
Apr 27 13:44:00 centos7.fenestros.loc mysqld_safe[30379]: 170427 13:44:00 mysqld_safe Logging to
'/var/log/mariadb/mariadb.log'.
Apr 27 13:44:00 centos7.fenestros.loc mysqld_safe[30379]: 170427 13:44:00 mysqld_safe Starting mysqld daemon with
databases from /var/lib/mysql
Apr 27 13:44:02 centos7.fenestros.loc systemd[1]: Started MariaDB database server.
```

Configuration

Lors de l'installation MariaDB n'a pas de mot de passe attribué à l'utilisateur root. Créez donc le mot de passe **fenestros** puis testez la connexion à MariaDB :

```
[root@centos7 ~]# mysqladmin -u root password fenestros
[root@centos7 ~]# mysql
ERROR 1045 (28000): Access denied for user 'root'@'localhost' (using password: NO)
[root@centos7 ~]# mysql -uroot -p
Enter password: fenestros
```

```
Welcome to the MariaDB monitor.  Commands end with ; or \g.
Your MariaDB connection id is 4
Server version: 5.5.64-MariaDB MariaDB Server

Copyright (c) 2000, 2018, Oracle, MariaDB Corporation Ab and others.

Type 'help;' or '\h' for help. Type '\c' to clear the current input statement.

MariaDB [(none)]> exit
Bye
```

OpenLDAP

Présentation

LDAP est une abréviation de **L**ighweight **D**irectory **A**ccess **P**rotocol. Comme son nom indique, LDAP est un service d'**annuaire**.

Un service d'annuaire est une base de données spécialisée optimisée pour la consultation. Certains services d'annuaire peuvent être locaux tandis que d'autres sont appelés **distribués**. Un bon exemple d'une service d'annuaire distribué est le **DNS**.

L'installation du serveur OpenLDAP ne peut être réussie que dans le cas où le support pour les bases de données **BerkeleyDB** soit installé au préalable.

Installation

Pour installer le serveur OpenLDAP, utilisez la commande yum :

```
[root@centos7 ~]# yum install openldap-servers openldap-clients openldap
```

Ensuite démarrez votre serveur OpenLDAP :

```
[root@centos7 ~]# systemctl status slapd.service
● slapd.service - OpenLDAP Server Daemon
   Loaded: loaded (/usr/lib/systemd/system/slapd.service; disabled; vendor preset: disabled)
   Active: inactive (dead)
     Docs: man:slapd
           man:slapd-config
           man:slapd-hdb
           man:slapd-mdb
           file:///usr/share/doc/openldap-servers/guide.html
[root@centos7 ~]# systemctl enable slapd.service
Created symlink from /etc/systemd/system/multi-user.target.wants/slapd.service to
/usr/lib/systemd/system/slapd.service.
[root@centos7 ~]# systemctl start slapd.service
[root@centos7 ~]# systemctl status slapd.service
● slapd.service - OpenLDAP Server Daemon
   Loaded: loaded (/usr/lib/systemd/system/slapd.service; enabled; vendor preset: disabled)
   Active: active (running) since Thu 2017-04-27 13:47:27 CEST; 2s ago
     Docs: man:slapd
           man:slapd-config
           man:slapd-hdb
           man:slapd-mdb
           file:///usr/share/doc/openldap-servers/guide.html
  Process: 32228 ExecStart=/usr/sbin/slapd -u ldap -h ${SLAPD_URLS} $SLAPD_OPTIONS (code=exited,
status=0/SUCCESS)
  Process: 32214 ExecStartPre=/usr/libexec/openldap/check-config.sh (code=exited, status=0/SUCCESS)
 Main PID: 32230 (slapd)
   CGroup: /system.slice/slapd.service
           └─32230 /usr/sbin/slapd -u ldap -h ldapi:/// ldap:///

Apr 27 13:47:27 centos7.fenestros.loc systemd[1]: Starting OpenLDAP Server Daemon...
Apr 27 13:47:27 centos7.fenestros.loc runuser[32217]: pam_unix(runuser:session): session opened for user ldap by
(uid=0)
Apr 27 13:47:27 centos7.fenestros.loc slapd[32228]: @(#) $OpenLDAP: slapd 2.4.40 (Nov  6 2016 01:21:28) $
mockbuild@worker1.bsys.centos.org:/builddir/build/BUILD/openldap-2.4.40/openldap-2.4.4...rs/slapd
```

```
Apr 27 13:47:27 centos7.fenestros.loc slapd[32230]: hdb_db_open: warning - no DB_CONFIG file found in directory
/var/lib/ldap: (2).
                                                    Expect poor performance for suffix "dc=my-domain,dc=com".
Apr 27 13:47:27 centos7.fenestros.loc slapd[32230]: slapd starting
Apr 27 13:47:27 centos7.fenestros.loc systemd[1]: Started OpenLDAP Server Daemon.
Hint: Some lines were ellipsized, use -l to show in full.
```

SER303 - Configuration du serveur Tomcat 8

Contenu du Module

        - LAB #1 -Journalisation des Requêtes Client dans un Fichier Texte
        - LAB #2 -Journalisation des Requêtes Client dans une Base de Données
      - L'élément <Listener>
  - Le Fichier $CATALINA_HOME/conf/web.xml
  - Le Fichier $CATALINA_HOME/conf/tomcat-users.xml
  - Le Fichier $CATALINA_HOME/conf/catalina.policy
  - Configuration des Ressources
      - Portée des Ressources
      - Pools de Connexion
      - Sessions JavaMail
      - JavaBeans
      - Entrées D'Environnement

Architecture du Serveur

Le répertoire d'installation de Tomcat est défini par la valeur de la variable $CATALINA_HOME :

```
[root@centos7 ~]# echo $CATALINA_HOME
/usr/tomcat8
```

Avant de continuer, installez l'utilitaire **tree** :

```
[root@centos7 tomcat8]# yum install tree
```

L'arborescence du répertoire $CATALINA_HOME est :

```
[root@centos7 ~]# cd $CATALINA_HOME
[root@centos7 tomcat8]# ls
bin  conf  lib  LICENSE  logs  NOTICE  RELEASE-NOTES  RUNNING.txt  temp  webapps  work
[root@centos7 tomcat8]# tree | more
.
├── bin
```

```
│   ├── bootstrap.jar
│   ├── catalina.bat
│   ├── catalina.sh
│   ├── catalina-tasks.xml
│   ├── commons-daemon.jar
│   ├── commons-daemon-native.tar.gz
│   ├── configtest.bat
│   ├── configtest.sh
│   ├── daemon.sh
│   ├── digest.bat
│   ├── digest.sh
│   ├── setclasspath.bat
│   ├── setclasspath.sh
│   ├── shutdown.bat
│   ├── shutdown.sh
│   ├── startup.bat
│   ├── startup.sh
│   ├── tomcat-juli.jar
│   ├── tomcat-native.tar.gz
│   ├── tool-wrapper.bat
│   ├── tool-wrapper.sh
│   ├── version.bat
│   └── version.sh
├── conf
│   ├── Catalina
│   │   └── localhost
│   ├── catalina.policy
│   ├── catalina.properties
│   ├── context.xml
│   ├── logging.properties
│   ├── server.xml
│   ├── tomcat-users.xml
│   ├── tomcat-users.xsd
│   └── web.xml
```

```
├── lib
│   ├── annotations-api.jar
│   ├── catalina-ant.jar
│   ├── catalina-ha.jar
--More--
```

Le répertoire **bin** contient des scripts et fichiers nécessaires au démarrage du serveur.

```
[root@centos7 tomcat8]# cd bin
[root@centos7 bin]# tree
.
├── bootstrap.jar
├── catalina.bat
├── catalina.sh
├── catalina-tasks.xml
├── commons-daemon.jar
├── commons-daemon-native.tar.gz
├── configtest.bat
├── configtest.sh
├── daemon.sh
├── digest.bat
├── digest.sh
├── setclasspath.bat
├── setclasspath.sh
├── shutdown.bat
├── shutdown.sh
├── startup.bat
├── startup.sh
├── tomcat-juli.jar
├── tomcat-native.tar.gz
├── tool-wrapper.bat
├── tool-wrapper.sh
├── version.bat
└── version.sh
```

```
0 directories, 23 files
```

Le répertoire **conf** contient les quatre fichiers de configuration importants :

- **server.xml**,
- **tomcat-users.xml**,
- **web.xml**,
- **catalina.policy**.

```
[root@centos7 bin]# cd ../conf
[root@centos7 conf]# tree
.
├── Catalina
│   └── localhost
├── catalina.policy
├── catalina.properties
├── context.xml
├── logging.properties
├── server.xml
├── tomcat-users.xml
├── tomcat-users.xsd
└── web.xml

2 directories, 8 files
```

Le répertoire **libs** contient les bibliothèques Java (fichiers .jar) disponibles à toutes les applications ainsi qu'à Tomcat lui-même :

```
[root@centos7 conf]# cd ../lib/
[root@centos7 lib]# tree
.
├── annotations-api.jar
├── catalina-ant.jar
├── catalina-ha.jar
├── catalina.jar
```

```
├── catalina-storeconfig.jar
├── catalina-tribes.jar
├── ecj-4.5.jar
├── el-api.jar
├── jasper-el.jar
├── jasper.jar
├── jsp-api.jar
├── servlet-api.jar
├── tomcat-api.jar
├── tomcat-coyote.jar
├── tomcat-dbcp.jar
├── tomcat-i18n-es.jar
├── tomcat-i18n-fr.jar
├── tomcat-i18n-ja.jar
├── tomcat-jdbc.jar
├── tomcat-jni.jar
├── tomcat-util.jar
├── tomcat-util-scan.jar
├── tomcat-websocket.jar
└── websocket-api.jar

0 directories, 24 files
```

Le répertoire **logs** contient les fichiers de journalisation du serveur :

```
[root@centos7 lib]# cd ../logs
[root@centos7 logs]# tree
.
├── catalina.2017-04-27.log
├── catalina.out
├── host-manager.2017-04-27.log
├── localhost.2017-04-27.log
├── localhost_access_log.2017-04-27.txt
└── manager.2017-04-27.log
```

```
0 directories, 6 files
```

Le répertoire **webapps** contient les deux applications pour la gestion du serveur :

```
[root@centos7 logs]# cd ../webapps/
[root@centos7 webapps]# tree | more
.
├── docs
│   ├── aio.html
│   ├── api
│   │   └── index.html
│   ├── appdev
│   │   ├── build.xml.txt
│   │   ├── deployment.html
│   │   ├── index.html
│   │   ├── installation.html
│   │   ├── introduction.html
│   │   ├── processes.html
│   │   ├── sample
│   │   │   ├── build.xml
│   │   │   ├── docs
│   │   │   │   └── README.txt
│   │   │   ├── index.html
│   │   │   ├── sample.war
│   │   │   ├── src
│   │   │   │   └── mypackage
│   │   │   │       └── Hello.java
│   │   │   └── web
│   │   │       ├── hello.jsp
│   │   │       ├── images
│   │   │       │   └── tomcat.gif
│   │   │       ├── index.html
│   │   │       └── WEB-INF
│   │   │           └── web.xml
```

```
│   │   ├── source.html
│   │   └── web.xml.txt
│   ├── apr.html
│   ├── architecture
│   │   ├── index.html
│   │   ├── overview.html
│   │   ├── requestProcess
│   │   │   ├── authentication-process.png
│   │   │   └── request-process.png
│   │   ├── requestProcess.html
│   │   ├── startup
│   │   │   ├── serverStartup.pdf
--More--
```

Le répertoire **temp** est utilisé comme répertoire temporaire par les applications :

```
[root@centos7 webapps]# cd ../temp/
[root@centos7 temp]# tree
.
└── safeToDelete.tmp

0 directories, 1 file
```

Le répertoire **work** contient de sous-répertoires pour chaque application utilisés pour la transformation des pages JSP en classes Java :

```
[root@centos7 temp]# cd ../work
[root@centos7 work]# tree
.
└── Catalina
    └── localhost
        ├── docs
        ├── examples
        ├── host-manager
        ├── manager
```

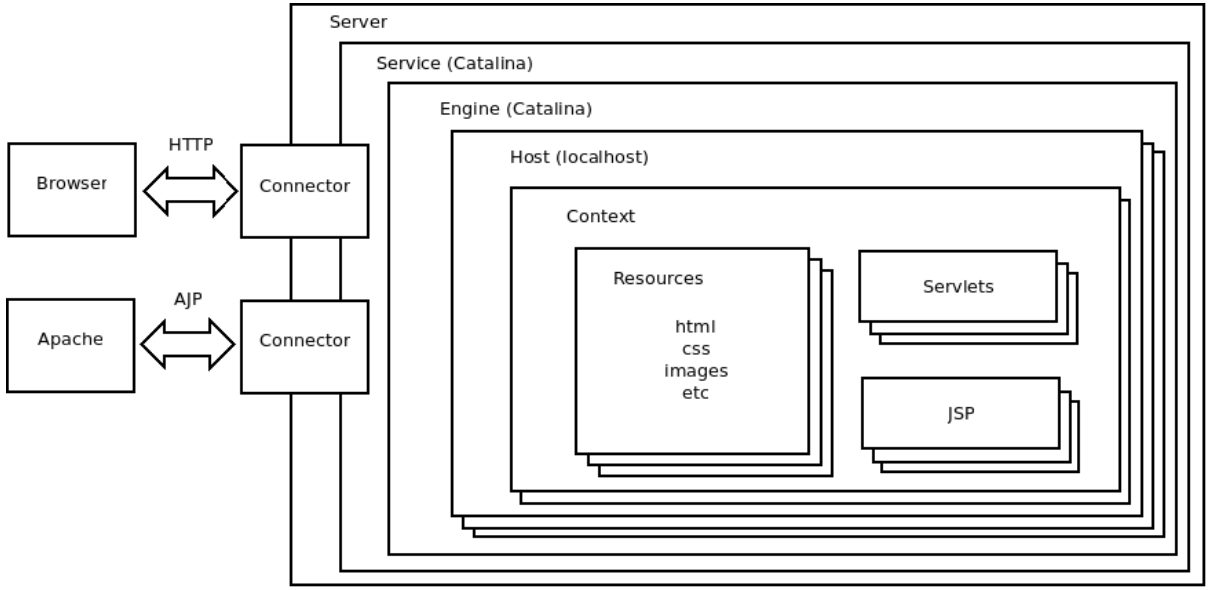
```
        └── ROOT
            └── org
                └── apache
                    └── jsp
                        ├── index_jsp.class
                        └── index_jsp.java

10 directories, 2 files
```

Fichiers de Configuration

Les composants de Tomcat sont appelés des **conteneurs**. Il en existe 5 :

- Server,
- Service,
- Engine,
- Host,
- Context.

L'organisation est imbriquée comme montre l'exemple suivant :



Cette organisation est détaillée dans le principal fichier de configuration de Tomcat : **$CATALINA_HOME/conf/server.xml**.

Le Fichier $CATALINA_HOME/conf/server.xml

A l'installation de Tomcat, ce fichier comporte les directives actives suivantes :

```
<?xml version='1.0' encoding='utf-8'?>
<Server port="8005" shutdown="SHUTDOWN">
  <Listener className="org.apache.catalina.startup.VersionLoggerListener" />
  <Listener className="org.apache.catalina.core.AprLifecycleListener" SSLEngine="on" />
  <Listener className="org.apache.catalina.core.JreMemoryLeakPreventionListener" />
  <Listener className="org.apache.catalina.mbeans.GlobalResourcesLifecycleListener" />
  <Listener className="org.apache.catalina.core.ThreadLocalLeakPreventionListener" />
  <GlobalNamingResources>
    <Resource name="UserDatabase" auth="Container"
              type="org.apache.catalina.UserDatabase"
              description="User database that can be updated and saved"
```

```
              factory="org.apache.catalina.users.MemoryUserDatabaseFactory"
              pathname="conf/tomcat-users.xml" />
  </GlobalNamingResources>
  <Service name="Catalina">
    <Connector port="8080" protocol="HTTP/1.1"
               connectionTimeout="20000"
               redirectPort="8443" />
    <Connector port="8009" protocol="AJP/1.3" redirectPort="8443" />
    <Engine name="Catalina" defaultHost="localhost">
      <Realm className="org.apache.catalina.realm.LockOutRealm">
        <Realm className="org.apache.catalina.realm.UserDatabaseRealm"
               resourceName="UserDatabase"/>
      </Realm>
      <Host name="localhost"  appBase="webapps"
            unpackWARs="true" autoDeploy="true">
        <Valve className="org.apache.catalina.valves.AccessLogValve" directory="logs"
               prefix="localhost_access_log" suffix=".txt"
               pattern="%h %l %u %t &quot;%r&quot; %s %b" />

      </Host>
    </Engine>
  </Service>
</Server>
```

Dans ce fichier, les éléments **obligatoires** sont démontrés par l'exemple suivant :

```
<Server ...>
    <Service ...>
        <Connector ...>
        </Connector ...>
        <Connector ...>
        </Connector ...>
                etc
        <Engine ...>
```

```
            <Host ...>
                <Context ...>
                </Context ...>
                <Context ...>
                </Context ...>
                <Context ...>
                                    etc
            </Host ...>
                        etc
        </Engine ...>
    </Service ...>
    <Service ...>
        <Connector ...>
        </Connector ...>
        <Connector ...>
        </Connector ...>
                etc
        <Engine ...>
            <Host ...>
                <Context ...>
                </Context ...>
                <Context ...>
                </Context ...>
                <Context ...>
                                    etc
            </Host ...>
                        etc
        </Engine ...>
    </Service ...>
        etc
</Server ...>
```

D'autres éléments peuvent aussi être employés. L'exemple suivant démontre leur positionnement dans l'organisation du fichier :

```
<Server ...>
    <Listener>
    </Listener>
    <GlobalNamingResources ...>
        <Environment ...>
        </Environment ...>
        <Resource ...>
        </Resource ...>
    </GlobalNamingResources...>
    <Service ...>
                <Executor...>
                </Executor>
        <Engine ...>
            <Logger ...>
            </Logger ...>
            <Realm ...>
            </Realm ...>
            <Valve ...>
            </Valve ...>
            <Listener ...>
            </Listener ...>
            <Host ...>
                <Alias>
                </Alias>
                <Logger ...>
                </Logger ...>
                <Realm ...>
                </Realm ...>
                <Valve ...>
                </Valve ...>
                <Listener ...>
                </Listener ...>
                <Context ...>
                    <Logger ...>
```

```
                    </Logger ...>
                    <Realm ...>
                    </Realm ...>
                    <Valve ...>
                    </Valve ...>
                    <Loader ...>
                    </Loader ...>
                    <Manager ...>
                        <Store ...>
                        </Store ...>
                    </Manager ...>
                    <Resource ...>
                    </Resource ...>
                    <ResourceLink ...>
                    </ResourceLink ...>
                </Context ...>
            </Host ...>
        </Engine ...>
    </Service ...>
</Server ...>
```

L'élément <Server>

Cet élément est la racine du fichier server.xml et comporte deux **attributs** :

- **port** - le port que Tomcat écoute pour une commande d'arrêt du serveur. Sa valeur par défaut est de **8005**,
- **shutdown** - la chaîne de caractères à envoyer au **port** afin que le serveur s'arrête. Sa valeur par défaut est **SHUTDOWN**.

<note warning> **Attention** - Afin d'éviter à ce qu'une personne malintentionnée puisse se connecter au port par défaut en utilisant telnet et envoyer la chaîne de caractères par défaut, il est impératif de modifier ses deux valeurs. </note>

L'élément <Service>

Cet élément permet d'associer des <Connector> et des <Executor> à un <Engine>. Le nom de chaque <Service> déclaré doit être unique.

L'attribut obligatoire est **name** :

Attribut	Description	Valeur par défaut	Valeur recommandée
name	Spécifie le nom du service	Catalina	**A Modifier**

L'élément <Connector>

Cet élément permet de mettre en place le transport de requêtes clients vers le moteur de servlet défini par l'élément <Engine> dans le même élément <Service>.

Il existe deux types selon le protocole utilisé :

- **HTTP** - le protocole HTTP/1.1 sur le port **8080**,
- **AJP** - le protocole AJP/1.3 sur le port **8009**.

Les attributs les plus importants de cet élément sont :

Attribut	Description	Valeur par défaut	Valeur recommandée
port	Spécifie le port d'écoute	8080/8009	8080/8009
address	Spécifie une adresse IP d'écoute	-	-
secure	Spécifie l'utilisation de HTTPS	false	-
redirectport	Spécifie le port vers lequel les requêtes HTTPS doivent être redirigées. L'attribut se trouve dans un Connector HTTPS.	-	-
disableUploadTimeout	Spécifie le temps maximum pour l'envoi de fichiers	false	-
enableLookups	Active la résolution des adresses IP en nom d'hôtes	true	**false**
proxyName	Spécifie le nom du proxy éventuel derrière lequel se trouve le serveur Tomcat	-	-
proxyPort	Spécifie le numéro du port du proxy éventuel derrière lequel se trouve le serveur Tomcat	-	-

Attribut	Description	Valeur par défaut	Valeur recommandée
maxThreads	Spécifie le nombre maximum de threads qui peut être créés dans un **pool**	200	200
acceptCount	Spécifie le nombre de requêtes qui sont mises en attente une fois la valeur de maxThreads ait été dépassée.	100	100
connectionTimeout	Spécifie le nombre de millisecondes qu'une requête puisse rester dans la file d'attente	60000 ms	60000 ms
executor	Spécifie le nom de l'élément <Executor> dont le <Connector> doit utiliser le pool	-	-

L'élément <Executor>

Pour traiter les requêtes entrantes, les <Connectors> utilisent des **threads** qui sont des sous-processus dans la machine virtuelle Java. Chaque thread est dédié à une connexion. Quand la connexion est terminée, le thread peut être ré-utilisé. Tomcat utilise un **pool** de threads pour éviter la création et la destruction inutiles de threads. Si les threads sont inactifs pour une période trop importante, le pool détruit les threads.

Les attributs les plus importants de cet élément sont :

Attribut	Description	Valeur par défaut	Valeur recommandée
maxThreads	Spécifie le nombre maximum de threads qui peut être créés dans un **pool**	200	200
maxSpareThreads	Spécifie le nombre maximum de threads inactifs qui peut être dans un **pool**	50	50
minSpareThreads	Spécifie le nombre minimum de threads inactifs qui peut être dans un **pool**	25	25
maxIdleTime	Spécifie le nombre de millisecondes du délai d'inactivité d'un thread avant sa déstruction	60000 ms	60000 ms
namePrefix	Spécifie un préfixe de nommage pour chaque thread du pool	-	-

L'élément <Engine>

Cet élément permet de mettre en place le transport de requêtes clients reçues des <Connector> vers les applications concernées.

Les attributs les plus importants de cet élément sont :

Attribut	Description	Valeur par défaut	Valeur recommandée
name	Spécifie le nom du moteur	Catalina	Catalina

Attribut	Description	Valeur par défaut	Valeur recommandée
defaultHost	Spécifie le nom d'hôte par défaut spécifiés par les élements <HOST>	-	-
jvmRoute	Spécifie les informations de l'**affinité de session** utilisées lors du clustering	-	-

L'élément <Host>

Cet élément permet de mettre en place un hôte unique, virtuel ou réel, dans le serveur.

Les attributs les plus importants de cet élément sont :

Attribut	Description	Valeur par défaut	Valeur recommandée
name	Spécifie le nom d'hôte	localhost	-
appBase	Spécifie le répertoire de stockage par défaut des applications	$CATALINA_HOME/webapps	$CATALINA_HOME/webapps
autoDeploy	Spécifie si les applications déposées dans $CATALINA_HOME/webapps sont à déployer automatiquement	true	**false**
deployOnStartup	Spécifie si les applications sont déployées automatiquement lors du démarrage du serveur	true	true
unpackWars	Spécifie si les archives compressées au format .war doivent être décompressées automatiquement	true	true
deployXML	Spécifie que le déploiement des applications peut être autorisé à partir des fichiers de contexte	true	true
workDir	Spécifie le répertoire de travail de l'application	-	-

L'élément <Context>

L'élément <Context> décrit une application déployée dans le serveur. Il existe deux types de <Context> :

- Les Contexts déclarés par un élément <Context>,
- Les Contexts déclarés automatiquement par le serveur.

Dans le premier des deux cas, la déclaration de l'élément peut se trouver soit dans :

- le fichier **server.xml**,

- le fichier de contexte XML de l'application concernée.

C'est ce deuxième cas qui est le plus souvent utilisé.

Les attributs les plus importants de cet élément sont :

Attribut	**Description**	**Valeur par défaut**	**Valeur recommandée**
docBase	Spécifie le chemin absolu ou relatif du répertoire des données ou du fichier .war de l'application	-	-
path	Spécifie le chemin de contexte de l'application	-	-
reloadable	Active une surveillance des répertoires /WEB-INF/lib et /WEB-INF/classes. Chaque modification apportée au contenu des deux répertoires est automatiquement prise en charge par le serveur	false	false
useNaming	Active un contexte de nommage JNDI permettant la recherche des ressources	true	true
swallowOutput	Spécifie que la sortie standard et la sortie d'erreur doivent être consignées dans le fichier journal de l'application	false	**true**
workDir	Spécifie le répertoire de travail de l'application en remplaçant la valeur de l'attribut dans l'élément <Host>	-	-
privileged	Autorise l'application d'utiliser les servlets du moteur Catalina (p.e. Manager)	false	false
cookies	Spécifie que les cookies sont utilisés pour implémenter les sessions HTTP	true	true
override	Spécifie que les valeurs des attributs du <Context> surchargent celles du <DefaultContext>	false	false

Le contexte par défaut <DefaultContext> se trouve dans le fichier **$CATALINA_HOME/conf/context.xml** :

```
<?xml version='1.0' encoding='utf-8'?>
<Context>

    <!-- Default set of monitored resources. If one of these changes, the    -->
    <!-- web application will be reloaded.                                   -->
    <WatchedResource>WEB-INF/web.xml</WatchedResource>
    <WatchedResource>${catalina.base}/conf/web.xml</WatchedResource>

    <!-- Uncomment this to disable session persistence across Tomcat restarts -->
    <!--
    <Manager pathname="" />
```

```
    -->

    <!-- Uncomment this to enable Comet connection tacking (provides events
         on session expiration as well as webapp lifecycle) -->
    <!--
    <Valve className="org.apache.catalina.valves.CometConnectionManagerValve" />
    -->
</Context>
```

L'élément <Realm>

Cet élément permet de gérer le système d'authentification. L'élément <Realm> peut être défini dans un des éléments suivants :

- <Engine> - le système d'authentification s'applique au moteur,
- <Host> - le système d'authentification s'applique au hôte,
- <Context> - le système d'authentification s'applique à l'application.

A noter qu'il ne peut être défini qu'une seule fois par élément et qu'il existe un héritage de l'élément enfant à partir de l'élément parent.

L'attribut fondamental de l'élément <Realm> est **className** qui prend une des valeurs suivantes :

Attribut	Valeur	Description
className	org.apache.catalina.realm.UserDatabaseRealm	Authentification utilisant le fichier $CATALINA_HOME/conf/tomcat-users.xml
className	org.apache.catalina.realm.JDBCRealm	Authentification utilisant une base de données
className	org.apache.catalina.realm.DataSourceRealm	Authentification utilisant une base de données
className	org.apache.catalina.realm.JNDIRealm	Authentification basée sur un annuaire LDAP
className	org.apache.catalina.realm.JAASRealm	Authentification basée sur l'API de sécurité Java (JAAS)
className	org.apache.catalina.realm.CombinedRealm	Authentification basée sur plusieurs Realms définis en tant qu'enfants

L'élément <Loader>

Les classes Java sont chargées dans un ordre spécifique par défaut :

- les classes dans le répertoire **/WEB-INF/classes**,
- les classes dans les fichiers .jar dans le répertoire **/WEB-INF/lib**,
- les classes rendues accessibles aux applications par le moteur de servlets.

L'élément facultatif <Loader> permet de modifier l'ordre par défaut pour une application donnée en chargeant d'abord les classes recherchées par les chargeurs parents de Tomcat.

Les attributs les plus importants de cet élément sont :

Attribut	**Description**	**Valeur par défaut**	**Valeur recommandée**
delegate	Permet de modifier l'ordre par défaut	false	false
reloadable	Active une surveillance des répertoires /WEB-INF/lib et /WEB-INF/classes. Chaque modification apportée au contenu des deux répertoires est automatiquement prise en charge par le serveur	Héritée de <Context>	-
checkInterval	Intervalle de temps de la vérification de l'attribut reloadable	15s	15s

L'élément <Manager>

Cet élément permet de configurer la gestion des sessions.

L'attribut fondamental de l'élément <Manager> est **className** qui prend une des valeurs suivantes :

Attribut	**Valeur**	**Description**
className	org.apache.catalina.session.StandardManager	Les sessions sont stockées dans un fichier
className	org.apache.catalina.session.PersistentManager	Les session sont stockées dans un entrepôt persistant tel une base de données

Les attributs communs aux deux valeurs de **className** sont :

Attribut	**Description**	**Valeur par défaut**	**Valeur recommandée**
maxActiveSessions	Le nombre maximum de sessions actives (-1 équivaut pas de limite)	-1	-1
distributable	Spécifie si les attributs de session peuvent être stockés dans un fichier ou une base de données	false	false

L'attribut spécifique à **org.apache.catalina.session.StandardManager** est :

Attribut	Description	Valeur par défaut	Valeur recommandée
pathname	Spécifie un chemin relatif ou absolu vers le fichier de stockage	SESSIONS.ser	SESSIONS.ser

Les attributs spécifiques à **org.apache.catalina.session.PersistentManager** sont :

Attribut	Description	Valeur par défaut	Valeur recommandée
saveOnRestart	Spécifie que les sessions soient sauvegardées quand Tomcat est arrêtée	true	true
maxIdleBackup	Spécifie l'intervalle de temps en secondes depuis le dernier accès à la session avant qu'elle soit éligible pour être persistée. La valeur de **-1** désactive cette fonctionnalité	-1	-1
maxIdleSwap	Spécifie l'intervalle de temps en secondes depuis le dernier accès à la session avant qu'elle soit persistée et supprimée de la mémoire. Cette valeur doit être supérieure ou égale à la valeur de **maxIdleBackup**. La valeur de **-1** désactive cette fonctionnalité	-1	-1

L'élément <Store>

Cet élément est un élément enfant de <Manager>. Il doit être spécifié si la valeur de **org.apache.catalina.session.PersistentManager** est **true**.

L'attribut fondamental de l'élément <Store> est **className** qui prend une des valeurs suivantes :

Attribut	Valeur	Description
className	org.apache.catalina.session.FileStore	L'entrepôt persistant est un fichier
className	org.apache.catalina.session.JDBCStore	L'entrepôt persistant est une base de données

L'élément <Valve>

L'élément <Valve> est à considérer comme un **filtre**.

L'élément <Valve> peut être défini dans un des éléments suivants :

- <Engine> - le filtre s'applique au moteur,

- <Host> - le filtre s'applique au hôte,
- <Context> - le filtre s'applique à l'application.

L'attribut fondamental de l'élément <Valve> est **className** qui prend une des valeurs suivantes :

Attribut	Valeur	Description
className	org.apache.catalina.valves.AccessLogValve	Génère un journal des accès au serveur dans un fichier
className	org.apache.catalina.valves.ExtendedAccessLogValve	Génère un journal des accès au serveur dans un fichier au format Extended Log File Format du W3C
className	org.apache.catalina.valves.JDBCAccessLogValve	Génère un journal des accès au serveur dans une base de données
className	org.apache.catalina.valves.RemoteAddrvalve	Applique une restriction des accès en fonction de l'IP du client
className	org.apache.catalina.valves.RemoteHostValve	Applique une restriction des accès en fonction du nom de la machine du client
className	org.apache.catalina.valves.SingleSignOn	Permet une authentification unique entre plusieurs applications
className	org.apache.catalina.valves.RemoteIPValve	Permet de récupérer l'adresse IP du client même si la requête est passée par un reverse proxy

Pour clarifier l'utilisation des filtres voici deux exemples :

Filtrage de l'adresse IP

```
<Valve ClassName="org.apache.catalina.valves.RemoteAddrvalve" allow="127.0.0.1, 10.*" />
```

Filtrage de nom de la machine du client

```
<Valve ClassName="org.apache.catalina.valves.RemoteHostValve" Deny="*.fenestros.com" />
```

LAB #1 -Journalisation des Requêtes Client dans un Fichier Texte

Tomcat propose deux formats ou *motifs* prédéfinis pour générer des journaux :

- **common**
  - %h %l %u %t ”%r” %s %b
- **combined**
  - %h %l %u %t “%r” %s %b “%{Referer}i” “%{User-Agent}i”

Dans les patterns ci-dessus, les significations des chaînes de formatage sont les suivantes :

Chaîne	Description
%h	Le nom d'hôte distant ou l'adresse IP si resolveHosts vaut false
%u	Le nom du compte de l'utilisateur distant s'il y a eu authentification
%t	L'heure de la réception de la requête par le serveur
%r	La première ligne de la requête
%s	Le statut de la dernière requête
%b	La taille de la réponse sans les entêtes HTTP. Le caractère **-** indique 0
%{Referer}i	Le nom du site d'où vient lé requête, contenu dans l'entête de la requête HTTP
%{User-Agent}i	Le nom du navigateur utilisé par le client, contenu dans l'entête de la requête HTTP

Pour mettre en place la journalisation personnalisée modifiez le fichier **$CATALINA_HOME/conf/server.xml** en commentant la section suivante :

```
...
         <!-- <Valve className="org.apache.catalina.valves.AccessLogValve" directory="logs"
              prefix="localhost_access_log" suffix=".txt"
              pattern="%h %l %u %t &amp;quot;%r&amp;quot; %s %b" /> -->

      </Host>
    </Engine>
  </Service>
```

et ajoutez la section suivante juste après :

```
...
        <Valve className="org.apache.catalina.valves.AccessLogValve"
        directory="logs" prefix="tomcat_access." suffix=".txt"
```

```
        resolveHosts="false" fileDateFormat="yyyy_MM_dd"
        pattern="%t - %a %H %s - %r"/>
...
```

Vous obtiendrez un résultat similaire au suivant :

```
...
        <!-- 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" directory="logs"
               prefix="localhost_access_log" suffix=".txt"
               pattern="%h %l %u %t &amp;quot;%r&amp;quot; %s %b" /> -->

        <Valve className="org.apache.catalina.valves.AccessLogValve"
        directory="logs" prefix="tomcat_access." suffix=".txt"
        resolveHosts="false" fileDateFormat="yyyy_MM_dd"
        pattern="%t - %a %H %s - %r"/>

      </Host>
    </Engine>
  </Service>
```

Dans cet exemple, on peut noter l'utilisation de plusieurs attributs supplémentaires :

Attribut	Description	Valeur par défaut	Valeur de l'exemple
directory	Le chemin absolu ou relatif vers le répertoire de sauvegarde des journaux	$CATALINA_HOME/logs	logs
prefix	Le nom du fichier suivi par un point	access_log.	tomcat_access.
suffix	L'extension du fichier, précédée par un point	vide	.txt
resolveHosts	Transforme les adresse IP en nom d'hôte en utilisant un serveur DNS	false	false
fileDateFormat	Spécifie le format de date utilisé pour nommé le fichier	-	yyyy_MM_dd

Dans ce cas, le fichier généré sera nommé **tomcat_access.yyyy_MM_dd.txt**.

Un dernier attribut peut également être présent. Il s'agit de l'attribut **rotatable**. Dans le cas où cet attribut porte la valeur **vrai**, les fichiers de journalisation subiront une rotation en fonction du **fileDateFormat** :

Format de fileDateFormat	Périodicité de la rotation
yyyy_MM_dd	Journalier
yyyy_MM	Hebdomadaire
yyyy	Annuelle

Dernièrement les significations des chaînes de formatage supplémentaires dans l'exemple sont les suivantes :

Chaîne	Description
%a	L'adresse IP distante
%H	Le protocole utilisé dans la requête

Redémarrez maintenant le serveur Tomcat :

```
[root@centos7 work]# cd ../bin
[root@centos7 bin]# ./shutdown.sh
Using CATALINA_BASE:   /usr/tomcat8
Using CATALINA_HOME:   /usr/tomcat8
Using CATALINA_TMPDIR: /usr/tomcat8/temp
Using JRE_HOME:        /usr/lib/jvm/jre-1.8.0-openjdk-1.8.0.232.b09-0.el7_7.x86_64
Using CLASSPATH:       /usr/tomcat8/bin/bootstrap.jar:/usr/tomcat8/bin/tomcat-juli.jar
[root@centos7 bin]# ./startup.sh
Using CATALINA_BASE:   /usr/tomcat8
Using CATALINA_HOME:   /usr/tomcat8
Using CATALINA_TMPDIR: /usr/tomcat8/temp
Using JRE_HOME:        /usr/lib/jvm/jre-1.8.0-openjdk-1.8.0.232.b09-0.el7_7.x86_64
Using CLASSPATH:       /usr/tomcat8/bin/bootstrap.jar:/usr/tomcat8/bin/tomcat-juli.jar
Tomcat started.
```

Consultez la page web http://www.i2tch.loc/docs/ afin de générer des traces dans le log puis consultez le répertoire **/$CATALINA_HOME/logs/** :

```
[root@centos7 bin]# lynx --dump http://www.i2tch.loc/docs/
   [1] Tomcat Home
   [2]The Apache Software Foundation

Apache Tomcat 8

   Version 8.0.36, Jun 9 2016

Links

     * [3]Docs Home
     * [4]FAQ
     * [5]User Comments
...
[root@centos7 bin]# ls /$CATALINA_HOME/logs/
catalina.2019-10-28.log      localhost.2019-10-29.log
catalina.2019-10-29.log      localhost_access_log.2019-10-28.txt
catalina.out                 localhost_access_log.2019-10-29.txt
host-manager.2019-10-28.log  manager.2019-10-28.log
host-manager.2019-10-29.log  manager.2019-10-29.log
localhost.2019-10-28.log     tomcat_access.2019_10_29.txt
```

Important : Notez la présence d'un fichier dont le nom est au format **tomcat_access.yyyy-MM-dd.txt**.

Consultez le contenu du fichier **tomcat_access.yyyy-MM-dd.txt**. Vous obtiendrez un résultat similaire à celui-ci :

```
[root@centos7 bin]# cat /$CATALINA_HOME/logs/tomcat_access.2019_10_29.txt
[29/Oct/2019:13:11:56 +0100] - 10.0.2.51 HTTP/1.0 200 - GET /docs/ HTTP/1.0
```

A Faire : Vérifiez que le résultat est conforme au *motif* spécifié dans le fichier **$CATALINA_HOME/conf/server.xml**.

LAB #2 -Journalisation des Requêtes Client dans une Base de Données

Pour pouvoir utiliser MariaDB pour consigner le journal de connexions, il convient d'abord de créer le fichier **tomcat.sql** contenant les lignes suivantes :

```
CREATE DATABASE `tomcat`;
USE `tomcat`;
CREATE TABLE `tomcat`.`AccessLog` (
    `id` INTEGER UNSIGNED NOT NULL AUTO_INCREMENT,
    `remoteHost` CHAR(15) NOT NULL DEFAULT '',
    `userName` CHAR(15),
    `timestamp` TIMESTAMP NOT NULL DEFAULT 0,
    `virtualHost` VARCHAR(64) NOT NULL DEFAULT '',
    `method` VARCHAR(8) NOT NULL DEFAULT '',
    `query` VARCHAR(255) NOT NULL DEFAULT '',
    `status` INTEGER UNSIGNED NOT NULL DEFAULT 0,
    `bytes` INTEGER UNSIGNED NOT NULL DEFAULT 0,
    `referer` VARCHAR(128),
    `userAgent` VARCHAR(128),
    PRIMARY KEY (`id`)
) ENGINE=InnoDB DEFAULT CHARACTER SET latin1;
```

Par exemple :

```
[root@centos7 bin]# vi tomcat.sql
[root@centos7 bin]# cat tomcat.sql
CREATE DATABASE `tomcat`;
USE `tomcat`;
CREATE TABLE `tomcat`.`AccessLog` (
    `id` INTEGER UNSIGNED NOT NULL AUTO_INCREMENT,
    `remoteHost` CHAR(15) NOT NULL DEFAULT '',
    `userName` CHAR(15),
    `timestamp` TIMESTAMP NOT NULL DEFAULT 0,
```

```
    `virtualHost` VARCHAR(64) NOT NULL DEFAULT '',
    `method` VARCHAR(8) NOT NULL DEFAULT '',
    `query` VARCHAR(255) NOT NULL DEFAULT '',
    `status` INTEGER UNSIGNED NOT NULL DEFAULT 0,
    `bytes` INTEGER UNSIGNED NOT NULL DEFAULT 0,
    `referer` VARCHAR(128),
    `userAgent` VARCHAR(128),
    PRIMARY KEY (`id`)
) ENGINE=InnoDB DEFAULT CHARACTER SET latin1;
```

Ce fichier contient les commandes SQL nécessaires pour créer la base de données MariaDB qui recevra les traces. Injectez donc les commandes SQL dans MariaDB :

```
[root@centos7 bin]# mysql -u root -p tomcat < tomcat.sql
Enter password:fenestros
[root@centos7 bin]# mysql -u root -p
Enter password:fenestros
Welcome to the MariaDB monitor.  Commands end with ; or \g.
Your MariaDB connection id is 6
Server version: 5.5.64-MariaDB MariaDB Server

Copyright (c) 2000, 2018, Oracle, MariaDB Corporation Ab and others.

Type 'help;' or '\h' for help. Type '\c' to clear the current input statement.

MariaDB [(none)]> SHOW DATABASES;
+--------------------+
| Database           |
+--------------------+
| information_schema |
| mysql              |
| performance_schema |
| test               |
| tomcat             |
```

```
+--------------------+
5 rows in set (0.00 sec)

MariaDB [(none)]> USE tomcat;
Reading table information for completion of table and column names
You can turn off this feature to get a quicker startup with -A

Database changed
MariaDB [tomcat]> SHOW TABLES;
+------------------+
| Tables_in_tomcat |
+------------------+
| AccessLog        |
+------------------+
1 row in set (0.00 sec)

MariaDB [tomcat]> DESCRIBE AccessLog;
+-------------+------------------+------+-----+---------------------+----------------+
| Field       | Type             | Null | Key | Default             | Extra          |
+-------------+------------------+------+-----+---------------------+----------------+
| id          | int(10) unsigned | NO   | PRI | NULL                | auto_increment |
| remoteHost  | char(15)         | NO   |     |                     |                |
| userName    | char(15)         | YES  |     | NULL                |                |
| timestamp   | timestamp        | NO   |     | 0000-00-00 00:00:00 |                |
| virtualHost | varchar(64)      | NO   |     |                     |                |
| method      | varchar(8)       | NO   |     |                     |                |
| query       | varchar(255)     | NO   |     |                     |                |
| status      | int(10) unsigned | NO   |     | 0                   |                |
| bytes       | int(10) unsigned | NO   |     | 0                   |                |
| referer     | varchar(128)     | YES  |     | NULL                |                |
| userAgent   | varchar(128)     | YES  |     | NULL                |                |
+-------------+------------------+------+-----+---------------------+----------------+
11 rows in set (0.00 sec)
```

```
MariaDB [tomcat]> exit
Bye
```

Téléchargez maintenant le pilote JDBC pour MySQL :

```
[root@centos7 bin]# wget http://dev.mysql.com/get/Downloads/Connector-J/mysql-connector-java-5.1.39.tar.gz
```

Décompressez-le

```
[root@centos7 bin]# tar xvf mysql-connector-java-5.1.39.tar.gz
...
```

Copiez le fichier **mysql-connector-java-5.1.39-bin.jar** dans le répertoire **$CATALINA_HOME/lib** :

```
[root@centos7 bin]# cp mysql-connector-java-5.1.39/mysql-connector-java-5.1.39-bin.jar $CATALINA_HOME/lib
```

Ajoutez maintenant la section suivante au fichier $CATALINA_HOME/conf/server.xml **à la place des lignes précédemment ajoutées** :

```
[root@centos7 bin]# vi $CATALINA_HOME/conf/server.xml
[root@centos7 bin]# tail -n25 $CATALINA_HOME/conf/server.xml
      <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" directory="logs"
               prefix="localhost_access_log" suffix=".txt"
```

```
               pattern="%h %l %u %t &amp;quot;%r&amp;quot; %s %b" /> -->

<Valve className="org.apache.catalina.valves.JDBCAccessLogValve"
        connectionURL="jdbc:mysql://localhost:3306/tomcat?user=root&amp;password=fenestros"
        driverName="com.mysql.jdbc.Driver" tableName="AccessLog"
        resolveHosts="false" pattern="common" />

      </Host>
    </Engine>
  </Service>
</Server>
```

Redémarrez le serveur Tomcat :

```
[root@centos7 bin]# ./shutdown.sh
Using CATALINA_BASE:   /usr/tomcat8
Using CATALINA_HOME:   /usr/tomcat8
Using CATALINA_TMPDIR: /usr/tomcat8/temp
Using JRE_HOME:        /usr/lib/jvm/jre-1.8.0-openjdk-1.8.0.232.b09-0.el7_7.x86_64
Using CLASSPATH:       /usr/tomcat8/bin/bootstrap.jar:/usr/tomcat8/bin/tomcat-juli.jar
[root@centos7 bin]# ./startup.sh
Using CATALINA_BASE:   /usr/tomcat8
Using CATALINA_HOME:   /usr/tomcat8
Using CATALINA_TMPDIR: /usr/tomcat8/temp
Using JRE_HOME:        /usr/lib/jvm/jre-1.8.0-openjdk-1.8.0.232.b09-0.el7_7.x86_64
Using CLASSPATH:       /usr/tomcat8/bin/bootstrap.jar:/usr/tomcat8/bin/tomcat-juli.jar
Tomcat started.
```

Rechargez la page **http://www.i2tch.loc/docs/**. Consultez ensuite le contenu de votre table **AccessLog** de la base de données **tomcat**. Vous obtiendrez un résultat similaire à l'exemple qui suit :

```
[root@centos7 bin]# mysql -u root -p
Enter password:
Welcome to the MariaDB monitor.  Commands end with ; or \g.
```

```
Your MariaDB connection id is 9
Server version: 5.5.64-MariaDB MariaDB Server

Copyright (c) 2000, 2018, Oracle, MariaDB Corporation Ab and others.

Type 'help;' or '\h' for help. Type '\c' to clear the current input statement.

MariaDB [(none)]> USE tomcat;
Reading table information for completion of table and column names
You can turn off this feature to get a quicker startup with -A

Database changed
MariaDB [tomcat]>
MariaDB [tomcat]> SELECT * from AccessLog;
+----+------------+----------+---------------------+-------------+--------+--------+--------+-------+---------+--
---------+
| id | remoteHost | userName | timestamp           | virtualHost | method | query  | status | bytes | referer |
userAgent |
+----+------------+----------+---------------------+-------------+--------+--------+--------+-------+---------+--
---------+
|  1 | 10.0.2.51  | NULL     | 2019-10-29 13:18:29 |             |        | /docs/ |    200 | 16580 | NULL    |
NULL      |
+----+------------+----------+---------------------+-------------+--------+--------+--------+-------+---------+--
---------+
1 row in set (0.00 sec)

MariaDB [tomcat]> exit
Bye
```

L'élément <Listener>

Cet élément permet de définir des écouteurs d’événements sur les éléments suivants :

- <Server>,
- <Engine>,
- <Host>,
- <Context>.

Le seul attribut de <Listener> est **className** qui prend une valeur différente selon les informations à collecter.

Cinq écouteurs par défaut sont détaillés dans le fichier **$CATALINA_HOME/conf/server.xml** :

```
<Server port="8005" shutdown="SHUTDOWN">
  <Listener className="org.apache.catalina.startup.VersionLoggerListener" />
  <Listener className="org.apache.catalina.core.AprLifecycleListener" SSLEngine="on" />
  <Listener className="org.apache.catalina.core.JreMemoryLeakPreventionListener" />
  <Listener className="org.apache.catalina.mbeans.GlobalResourcesLifecycleListener" />
  <Listener className="org.apache.catalina.core.ThreadLocalLeakPreventionListener" />
  <GlobalNamingResources>
...
```

$CATALINA_HOME/conf/web.xml

Ce fichier définit les paramètres de configuration communs à toutes les applications de Tomcat. Les directives actives de ce fichier sont :

```
<?xml version="1.0" encoding="ISO-8859-1"?>

<web-app xmlns="http://xmlns.jcp.org/xml/ns/javaee"
  xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
  xsi:schemaLocation="http://xmlns.jcp.org/xml/ns/javaee
                      http://xmlns.jcp.org/xml/ns/javaee/web-app_3_1.xsd"
  version="3.1">

    <servlet>
        <servlet-name>default</servlet-name>
        <servlet-class>org.apache.catalina.servlets.DefaultServlet</servlet-class>
```

```
        <init-param>
            <param-name>debug</param-name>
            <param-value>0</param-value>
        </init-param>
        <init-param>
            <param-name>listings</param-name>
            <param-value>false</param-value>
        </init-param>
        <load-on-startup>1</load-on-startup>
    </servlet>

    <servlet>
        <servlet-name>jsp</servlet-name>
        <servlet-class>org.apache.jasper.servlet.JspServlet</servlet-class>
        <init-param>
            <param-name>fork</param-name>
            <param-value>false</param-value>
        </init-param>
        <init-param>
            <param-name>xpoweredBy</param-name>
            <param-value>false</param-value>
        </init-param>
        <load-on-startup>3</load-on-startup>
    </servlet>

    <servlet-mapping>
        <servlet-name>default</servlet-name>
        <url-pattern>/</url-pattern>
    </servlet-mapping>
    <servlet-mapping>
        <servlet-name>jsp</servlet-name>
        <url-pattern>*.jsp</url-pattern>
        <url-pattern>*.jspx</url-pattern>
    </servlet-mapping>
```

```
    <session-config>
        <session-timeout>30</session-timeout>
    </session-config>

    <mime-mapping>
        <extension>123</extension>
        <mime-type>application/vnd.lotus-1-2-3</mime-type>
    </mime-mapping>
    <mime-mapping>
        <extension>3dml</extension>
        <mime-type>text/vnd.in3d.3dml</mime-type>
    ...
    <mime-mapping>
        <extension>xht</extension>
        <mime-type>application/xhtml+xml</mime-type>
    </mime-mapping>
    <mime-mapping>
        <extension>xhtml</extension>
        <mime-type>application/xhtml+xml</mime-type>
    </mime-mapping>
    <mime-mapping>
        <extension>zir</extension>
        <mime-type>application/vnd.zul</mime-type>
    </mime-mapping>
    <mime-mapping>
        <extension>zirz</extension>
        <mime-type>application/vnd.zul</mime-type>
    </mime-mapping>
    <mime-mapping>
        <extension>zmm</extension>
        <mime-type>application/vnd.handheld-entertainment+xml</mime-type>
    </mime-mapping>

    <welcome-file-list>
```

```
        <welcome-file>index.html</welcome-file>
        <welcome-file>index.htm</welcome-file>
        <welcome-file>index.jsp</welcome-file>
    </welcome-file-list>
</web-app>
```

Dans ce fichier, on peut constater la présence de plusieurs sections :

- <servlet-name>default</servlet-name>,
  - Cette section contient un paramètre **listings** qui permet l'affichage du contenu d'un répertoire dans le cas où aucune page d'accueil soit présente,
- <servlet-name>jsp</servlet-name>,
  - Cette section est responsable de la transformation des pages JSP en servlets,
- <servlet-mapping>,
  - Cette section définit les servlets responsable en fonction des extensions de fichiers,
- <session-config>,
  - Cette section définit le timeout par défaut. La valeur est en minutes.
- <mime-mapping>,
  - Ces sections définissent les types de fichiers et leurs extensions,
- <welcome-file-list>,
  - Cette section définit les pages d'accueil par défaut.

$CATALINA_HOME/conf/tomcat-users.xml

Ce fichier est utilisé par le mécanisme d'authentification par défaut de Tomcat et contient des entrées de type *nom d'utilisateur*, *mot de passe* et *rôle* :

```
<?xml version='1.0' encoding='utf-8'?>
<tomcat-users xmlns="http://tomcat.apache.org/xml"
              xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
              xsi:schemaLocation="http://tomcat.apache.org/xml tomcat-users.xsd"
              version="1.0">
<!--
  <role rolename="tomcat"/>
```

```
  <role rolename="role1"/>
  <user username="tomcat" password="<must-be-changed>" roles="tomcat"/>
  <user username="both" password="<must-be-changed>" roles="tomcat,role1"/>
  <user username="role1" password="<must-be-changed>" roles="role1"/>
-->
</tomcat-users>
```

Depuis la version 7 de Tomcat, l'application **manager** un des rôles suivants :

- **manager-gui**,
  - permet l'accès au manager en mode HTML graphique,
- **manager-script**,
  - permet l'accès au manager en mode texte,
- **manager-jmx**,
  - permet uniquement l'accès aux pages de supervision JMX et d'état du serveur,
- **manager-status**,
  - permet uniquement l'accès à la page d'état du serveur.

Afin donc de pouvoir utiliser l'application manager en mode texte, **remplacez** le contenu du fichier **$CATALINA_HOME/conf/tomcat-users.xml** avec le contenu suivant :

```
[root@centos7 bin]# vi $CATALINA_HOME/conf/tomcat-users.xml
[root@centos7 bin]# cat $CATALINA_HOME/conf/tomcat-users.xml
<?xml version='1.0' encoding='utf-8'?>
<tomcat-users xmlns="http://tomcat.apache.org/xml"
              xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
              xsi:schemaLocation="http://tomcat.apache.org/xml tomcat-users.xsd"
              version="1.0">
  <role rolename="tomcat"/>
  <role rolename="role1"/>
  <role rolename="manager-script"/>
  <user username="tomcat" password="tomcat" roles="tomcat"/>
  <user username="both" password="tomcat" roles="tomcat,role1"/>
  <user username="role1" password="tomcat" roles="role1"/>
```

```
  <user username="admin" password="fenestros" roles="manager-script"/>
</tomcat-users>
```

Arrêtez puis démarrez le serveur Tomcat :

```
[root@centos7 bin]# ./shutdown.sh
Using CATALINA_BASE:   /usr/tomcat8
Using CATALINA_HOME:   /usr/tomcat8
Using CATALINA_TMPDIR: /usr/tomcat8/temp
Using JRE_HOME:        /usr/lib/jvm/jre-1.8.0-openjdk-1.8.0.232.b09-0.el7_7.x86_64
Using CLASSPATH:       /usr/tomcat8/bin/bootstrap.jar:/usr/tomcat8/bin/tomcat-juli.jar
[root@centos7 bin]# ./startup.sh
Using CATALINA_BASE:   /usr/tomcat8
Using CATALINA_HOME:   /usr/tomcat8
Using CATALINA_TMPDIR: /usr/tomcat8/temp
Using JRE_HOME:        /usr/lib/jvm/jre-1.8.0-openjdk-1.8.0.232.b09-0.el7_7.x86_64
Using CLASSPATH:       /usr/tomcat8/bin/bootstrap.jar:/usr/tomcat8/bin/tomcat-juli.jar
Tomcat started.
```

$CATALINA_HOME/conf/catalina.policy

Par défaut, Tomcat est démarré sans **Gestionnaire de Sécurité** (*SecurityManager*). Dans ce cas, les applications peuvent faire tout ce qu'elles veulent. Dans le cas où on souhaite démarrer Tomcat en utilisant le Gestionnaire de Sécurité, il convient de lancer Tomcat avec le script **startup.sh** en stipulant l'argument **security**. Dans ce cas, tout ce qui n'est pas autorisé dans le fichier **catalina.policy** est interdit.

```
// Licensed to the Apache Software Foundation (ASF) under one or more
// contributor license agreements.  See the NOTICE file distributed with
// this work for additional information regarding copyright ownership.
// The ASF licenses this file to You under the Apache License, Version 2.0
// (the "License"); you may not use this file except in compliance with
// the License.  You may obtain a copy of the License at
//
//     http://www.apache.org/licenses/LICENSE-2.0
```

```
//
// Unless required by applicable law or agreed to in writing, software
// distributed under the License is distributed on an "AS IS" BASIS,
// WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
// See the License for the specific language governing permissions and
// limitations under the License.

// ============================================================================
// catalina.policy - Security Policy Permissions for Tomcat
//
// This file contains a default set of security policies to be enforced (by the
// JVM) when Catalina is executed with the "-security" option.  In addition
// to the permissions granted here, the following additional permissions are
// granted to each web application:
//
// * Read access to the web application's document root directory
// * Read, write and delete access to the web application's working directory
// ============================================================================


// ========== SYSTEM CODE PERMISSIONS =========================================


// These permissions apply to javac
grant codeBase "file:${java.home}/lib/-" {
        permission java.security.AllPermission;
};

// These permissions apply to all shared system extensions
grant codeBase "file:${java.home}/jre/lib/ext/-" {
        permission java.security.AllPermission;
};

// These permissions apply to javac when ${java.home] points at $JAVA_HOME/jre
```

```
grant codeBase "file:${java.home}/../lib/-" {
        permission java.security.AllPermission;
};

// These permissions apply to all shared system extensions when
// ${java.home} points at $JAVA_HOME/jre
grant codeBase "file:${java.home}/lib/ext/-" {
        permission java.security.AllPermission;
};


// ========== CATALINA CODE PERMISSIONS =======================================


// These permissions apply to the daemon code
grant codeBase "file:${catalina.home}/bin/commons-daemon.jar" {
        permission java.security.AllPermission;
};

// These permissions apply to the logging API
// Note: If tomcat-juli.jar is in ${catalina.base} and not in ${catalina.home},
// update this section accordingly.
//  grant codeBase "file:${catalina.base}/bin/tomcat-juli.jar" {..}
grant codeBase "file:${catalina.home}/bin/tomcat-juli.jar" {
        permission java.io.FilePermission
         "${java.home}${file.separator}lib${file.separator}logging.properties", "read";

        permission java.io.FilePermission
         "${catalina.base}${file.separator}conf${file.separator}logging.properties", "read";
        permission java.io.FilePermission
         "${catalina.base}${file.separator}logs", "read, write";
        permission java.io.FilePermission
         "${catalina.base}${file.separator}logs${file.separator}*", "read, write";
```

```
        permission java.lang.RuntimePermission "shutdownHooks";
        permission java.lang.RuntimePermission "getClassLoader";
        permission java.lang.RuntimePermission "setContextClassLoader";

        permission java.lang.management.ManagementPermission "monitor";

        permission java.util.logging.LoggingPermission "control";

        permission java.util.PropertyPermission "java.util.logging.config.class", "read";
        permission java.util.PropertyPermission "java.util.logging.config.file", "read";
        permission java.util.PropertyPermission "org.apache.juli.AsyncLoggerPollInterval", "read";
        permission java.util.PropertyPermission "org.apache.juli.AsyncMaxRecordCount", "read";
        permission java.util.PropertyPermission "org.apache.juli.AsyncOverflowDropType", "read";
        permission java.util.PropertyPermission "org.apache.juli.ClassLoaderLogManager.debug", "read";
        permission java.util.PropertyPermission "catalina.base", "read";

        // Note: To enable per context logging configuration, permit read access to
        // the appropriate file. Be sure that the logging configuration is
        // secure before enabling such access.
        // E.g. for the examples web application (uncomment and unwrap
        // the following to be on a single line):
        // permission java.io.FilePermission "${catalina.base}${file.separator}
        //  webapps${file.separator}examples${file.separator}WEB-INF
        //  ${file.separator}classes${file.separator}logging.properties", "read";
};

// These permissions apply to the server startup code
grant codeBase "file:${catalina.home}/bin/bootstrap.jar" {
        permission java.security.AllPermission;
};

// These permissions apply to the servlet API classes
// and those that are shared across all class loaders
// located in the "lib" directory
```

```
grant codeBase "file:${catalina.home}/lib/-" {
        permission java.security.AllPermission;
};


// If using a per instance lib directory, i.e. ${catalina.base}/lib,
// then the following permission will need to be uncommented
// grant codeBase "file:${catalina.base}/lib/-" {
//         permission java.security.AllPermission;
// };


// ========== WEB APPLICATION PERMISSIONS =====================================


// These permissions are granted by default to all web applications
// In addition, a web application will be given a read FilePermission
// for all files and directories in its document root.
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";

    // OS Specific properties to allow read access
    permission java.util.PropertyPermission "os.name", "read";
    permission java.util.PropertyPermission "os.version", "read";
    permission java.util.PropertyPermission "os.arch", "read";
    permission java.util.PropertyPermission "file.separator", "read";
    permission java.util.PropertyPermission "path.separator", "read";
    permission java.util.PropertyPermission "line.separator", "read";

    // JVM properties to allow read access
```

```
    permission java.util.PropertyPermission "java.version", "read";
    permission java.util.PropertyPermission "java.vendor", "read";
    permission java.util.PropertyPermission "java.vendor.url", "read";
    permission java.util.PropertyPermission "java.class.version", "read";
    permission java.util.PropertyPermission "java.specification.version", "read";
    permission java.util.PropertyPermission "java.specification.vendor", "read";
    permission java.util.PropertyPermission "java.specification.name", "read";

    permission java.util.PropertyPermission "java.vm.specification.version", "read";
    permission java.util.PropertyPermission "java.vm.specification.vendor", "read";
    permission java.util.PropertyPermission "java.vm.specification.name", "read";
    permission java.util.PropertyPermission "java.vm.version", "read";
    permission java.util.PropertyPermission "java.vm.vendor", "read";
    permission java.util.PropertyPermission "java.vm.name", "read";

    // Required for OpenJMX
    permission java.lang.RuntimePermission "getAttribute";

    // Allow read of JAXP compliant XML parser debug
    permission java.util.PropertyPermission "jaxp.debug", "read";

    // All JSPs need to be able to read this package
    permission java.lang.RuntimePermission "accessClassInPackage.org.apache.tomcat";

    // Precompiled JSPs need access to these packages.
    permission java.lang.RuntimePermission "accessClassInPackage.org.apache.jasper.el";
    permission java.lang.RuntimePermission "accessClassInPackage.org.apache.jasper.runtime";
    permission java.lang.RuntimePermission
     "accessClassInPackage.org.apache.jasper.runtime.*";

    // Precompiled JSPs need access to these system properties.
    permission java.util.PropertyPermission
     "org.apache.jasper.runtime.BodyContentImpl.LIMIT_BUFFER", "read";
    permission java.util.PropertyPermission
```

```
     "org.apache.el.parser.COERCE_TO_ZERO", "read";

    // The cookie code needs these.
    permission java.util.PropertyPermission
     "org.apache.catalina.STRICT_SERVLET_COMPLIANCE", "read";
    permission java.util.PropertyPermission
     "org.apache.tomcat.util.http.ServerCookie.STRICT_NAMING", "read";
    permission java.util.PropertyPermission
     "org.apache.tomcat.util.http.ServerCookie.FWD_SLASH_IS_SEPARATOR", "read";

    // Applications using Comet need to be able to access this package
    permission java.lang.RuntimePermission "accessClassInPackage.org.apache.catalina.comet";

    // Applications using WebSocket need to be able to access these packages
    permission java.lang.RuntimePermission "accessClassInPackage.org.apache.tomcat.websocket";
    permission java.lang.RuntimePermission "accessClassInPackage.org.apache.tomcat.websocket.server";
};


// The Manager application needs access to the following packages to support the
// session display functionality. These settings support the following
// configurations:
// - default CATALINA_HOME == CATALINA_BASE
// - CATALINA_HOME != CATALINA_BASE, per instance Manager in CATALINA_BASE
// - CATALINA_HOME != CATALINA_BASE, shared Manager in CATALINA_HOME
grant codeBase "file:${catalina.base}/webapps/manager/-" {
    permission java.lang.RuntimePermission "accessClassInPackage.org.apache.catalina";
    permission java.lang.RuntimePermission "accessClassInPackage.org.apache.catalina.ha.session";
    permission java.lang.RuntimePermission "accessClassInPackage.org.apache.catalina.manager";
    permission java.lang.RuntimePermission "accessClassInPackage.org.apache.catalina.manager.util";
    permission java.lang.RuntimePermission "accessClassInPackage.org.apache.catalina.util";
};
grant codeBase "file:${catalina.home}/webapps/manager/-" {
    permission java.lang.RuntimePermission "accessClassInPackage.org.apache.catalina";
```

```
    permission java.lang.RuntimePermission "accessClassInPackage.org.apache.catalina.ha.session";
    permission java.lang.RuntimePermission "accessClassInPackage.org.apache.catalina.manager";
    permission java.lang.RuntimePermission "accessClassInPackage.org.apache.catalina.manager.util";
    permission java.lang.RuntimePermission "accessClassInPackage.org.apache.catalina.util";
};

// You can assign additional permissions to particular web applications by
// adding additional "grant" entries here, based on the code base for that
// application, /WEB-INF/classes/, or /WEB-INF/lib/ jar files.
//
// Different permissions can be granted to JSP pages, classes loaded from
// the /WEB-INF/classes/ directory, all jar files in the /WEB-INF/lib/
// directory, or even to individual jar files in the /WEB-INF/lib/ directory.
//
// For instance, assume that the standard "examples" application
// included a JDBC driver that needed to establish a network connection to the
// corresponding database and used the scrape taglib to get the weather from
// the NOAA web server.  You might create a "grant" entries like this:
//
// The permissions granted to the context root directory apply to JSP pages.
// grant codeBase "file:${catalina.base}/webapps/examples/-" {
//      permission java.net.SocketPermission "dbhost.mycompany.com:5432", "connect";
//      permission java.net.SocketPermission "*.noaa.gov:80", "connect";
// };
//
// The permissions granted to the context WEB-INF/classes directory
// grant codeBase "file:${catalina.base}/webapps/examples/WEB-INF/classes/-" {
// };
//
// The permission granted to your JDBC driver
// grant codeBase "jar:file:${catalina.base}/webapps/examples/WEB-INF/lib/driver.jar!/-" {
//      permission java.net.SocketPermission "dbhost.mycompany.com:5432", "connect";
// };
// The permission granted to the scrape taglib
```

```
// grant codeBase "jar:file:${catalina.base}/webapps/examples/WEB-INF/lib/scrape.jar!/-" {
//      permission java.net.SocketPermission "*.noaa.gov:80", "connect";
// };
```

Configuration des Ressources

Les ressources disponibles à Tomcat sont configurées en leur donnant un **nom JNDI**. L'API JNDI est un service de nommage sous forme d'une arborescence de ressources.

Portée des Ressources

Une ressource peut être déclarée à trois endroits différents. Selon cet endroit, sa portée est différente :

- dans l'élément <Context>, elle est alors disponible à l'application uniquement,
- dans le contexte par défaut, elle est alors disponible à toutes les applications d'un hôte,
- dans <GlobalNamingResources>, elle est alors disponible au serveur.

Une ressource est déclarée avec l'élément **<Resource>**. Elle :

- est immédiatement disponible à l'application si elle est déclarée dans un <Context>,
- est immédiatement disponible aux applications si elle est déclarée dans le contexte par défaut,
- nécessite d'être appelée par un élément **<ResourceLink>** si elle est déclarée dans <GlobalNamingResources>.

Les attributs de <Resource> sont :

Attribut	**Description**	**Valeur par défaut**	**Valeur recommandée**
name	Spécifie le nom JNDI	-	-
type	Spécifie le type de données java de la ressource	-	-
description	Une description de la ressource	-	-
auth	Le type d'authentification à la ressource	-	-

L'attribut **auth** peut avoir plusieurs valeurs différentes en fonction de l'emplacement de la configuration de l'authentification :

- Container - la configuration est dans le serveur,
- Application - la configuration est dans l'application.

L'**exemple** suivant, issu du fichier server.xml démontre la syntaxe :

```
<GlobalNamingResources>
  <Resource name="UserDatabase" auth="Container"
            type="org.apache.catalina.UserDatabase"
            description="User database that can be updated and saved"
            factory="org.apache.catalina.users.MemoryUserDatabaseFactory"
            pathname="conf/tomcat-users.xml" />
</GlobalNamingResources>
```

Pools de Connexion

Pour augmenter le temps de réponse aux requêtes multiples de connexions JDBC, JEE utilise un système de pool de connexions.

Un pool de connexion est :

- créé par le serveur et non par l'application,
- est utilisable par plusieurs applications en même temps,
- ré-utilise des connexions non-utilisées,
- basé sur la bibliothèque **commons-dbcp**,
- est défini dans un élément **<Resource>**.

Sessions JavaMail

JavaMail :

- est l'API de messagerie pour l'envoi et la réception d'emails,

- supporte les protocoles SMTP, POP3 et IMAP4,
- nécessite l'API **JAF** (*Java Activation Framework*),
- doit être téléchargé car il n'est pas inclus en standard dans Tomcat,
- est défini dans un élément **<Resource>**,

Pour activer JavaMail, il convient de copier les fichiers **mail.jar** et **activation.jar** dans le répertoire **$CATALINA_HOME/lib**.

JavaBeans

Un **JavaBean** :

- est un objet contenant :
  - des données,
  - des fonctions, appelées **méthodes**,
- est un conteneur de différentes données regroupées, par exemple, nom, prénom, adresse etc,
- est un conteneur de méthodes pour avoir accès aux données,
- est déclaré dans un élément **<Resource>**,
- nécessite un **<Factory>** pour se fabriquer,

Le <Factory> par défaut est **org.apache.naming.factory.BeanFactory**.

Entrées D'Environnement

Une **Entrée d'environnement** est à considérer comme une variable. Une Entrée d'environnement :

- se déclare par un élément <Environment> au même niveau que <Resource> et contient un élément <type> qui peut être :
  - java.lang.Boolean,
  - java.lang.Byte,
  - java.lang.Character,
  - java.lang.Double,
  - java.lang.Float,
  - java.lang.Integer,

- java.lang.Long,
- java.lang.Short,
- java.lang.String.

SER304 - Déploiement et Gestion des Applications

Contenu du Module

Déployer une application

Il existe plusieurs méthodes pour déployer une application sous Tomcat.

Déploiement Automatique

Dans ce cas, toute application dans le répertoire **$CATALINA_HOME/webapps** est automatiquement déployer si l'attribut **autoDeploy** est vrai dans l'élément **<Host>** du fichier **$CATALINA_HOME/conf/server.xml**.

L'Element Context

Le contexte d'une application, nécessaire pour son déploiement, est automatiquement créé quand une application est copiée dans le répertoire **$CATALINA_HOME/webapps**. Dans le cas où on souhaite déployer une application en dehors de ce répertoire, il convient d'utiliser l'élément **<Context>** dans le fichier **$CATALINA_HOME/conf/server.xml**.

L'élément prend la forme suivante :

<Context path=“/demo” docBase=“/un/autre/répertoire” />

Déploiement avec XML

Dans le cas précédent, le fichier **$CATALINA_HOME/conf/server.xml** ayant été modifié, il est nécessaire de re-démarrer le serveur.

Afin d'éviter ceci, l'élément <Context> peut être défini dans un fichier XML au nom de l'application. Ce fichier est à copier dans le répertoire **$CATALINA_HOME/conf/Catalina/localhost**.

L'élément prend la forme suivante :

<Context path=“/demo” docBase=“/un/autre/répertoire” />

Dans le cas où, l'attribut **autoDeploy** est vrai, l'application sera déployer automatiquement dès la détection de ce fichier.

Application Manager de Tomcat

L'**Application Manager** de Tomcat est utilisable à partir de trois interfaces :

- l'interface texte,
- l'interface HTML,
- l'interface ANT.

Cet outil permet de :

- déployer une application,
- obtenir la liste des applications actives,
- recharger une application,
- obtenir d'informations sur les ressources JNDI,
- démarrer une application,
- arrêter une application,
- supprimer un application.

L'interface Texte

La syntaxe des commandes est la suivante :

```
http://<hote>:<port>/manager/<commandes>?<paramètres>
```

Les commandes admises sont :

- list,
- deploy,
- start,
- stop,

- install,
- remove,
- undeploy,
- reload,
- serverinfo,
- roles,
- sessions,
- resources,
- status,
- jmxproxy.

Voici des exemples des commandes les plus intéressantes :

list

Saisissez la commande suivante :

```
[root@centos7 bin]# lynx --dump -auth=admin:fenestros http://www.i2tch.loc:8080/manager/text/list
OK - Listed applications for virtual host localhost
/:running:0:ROOT
/examples:running:0:examples
/host-manager:running:0:host-manager
/manager:running:0:manager
/docs:running:0:docs
```

Le format de chaque ligne est :

contexte:état:sessions:docBase

deploy

Saisissez maintenant la commande suivante :

```
[root@centos7 bin]# lynx --dump -auth admin:fenestros
"http://www.i2tch.loc:8080/manager/text/deploy?path=/sample&war=file:/usr/tomcat8/webapps/docs/appdev/sample/sample.war&update=true"
OK - Deployed application at context path /sample
```

> **Important** : Notez l'utilisation de **&update=true**. Cette option spécifie à la commande deploy que si l'application existe déjà dans le serveur le manager doit d'abord la supprimer pour ensuite l'installer de nouveau.

Notez la création du répertoire $CATALINA_HOME/webapps/sample :

```
[root@centos7 bin]# ls ../webapps/
docs  examples  host-manager  manager  ROOT  sample  sample.war
```

start

Saisissez maintenant la commande suivante :

```
[root@centos7 bin]# lynx --dump -auth admin:fenestros "http://www.i2tch.loc:8080/manager/text/start?path=/sample"
OK - Started application at context path /sample
```

stop

Saisissez maintenant la commande suivante :

```
[root@centos7 bin]# lynx --dump -auth admin:fenestros "http://www.i2tch.loc:8080/manager/text/stop?path=/sample"
OK - Stopped application at context path /sample
```

reload

Saisissez les commandes suivantes :

```
[root@centos7 bin]# lynx --dump -auth admin:fenestros "http://www.i2tch.loc:8080/manager/text/start?path=/sample"
OK - Started application at context path /sample
[root@centos7 bin]#
[root@centos7 bin]# lynx --dump -auth admin:fenestros
"http://www.i2tch.loc:8080/manager/text/reload?path=/sample"
OK - Reloaded application at context path /sample
```

undeploy

Saisissez maintenant la commande suivante :

```
[root@centos7 bin]# lynx --dump -auth admin:fenestros
"http://www.i2tch.loc:8080/manager/text/undeploy?path=/sample"
OK - Undeployed application at context path /sample
```

Notez la suppression du répertoire $CATALINA_HOME/webapps/sample :

```
[root@centos7 bin]# ls ../webapps/
docs  examples  host-manager  manager  ROOT
```

resources

Saisissez la commande suivante :

```
[root@centos7 bin]# lynx --dump -auth admin:fenestros "http://www.i2tch.loc:8080/manager/text/resources"
OK - Listed global resources of all types
```

```
UserDatabase:org.apache.catalina.users.MemoryUserDatabase
```

Notez qu'il est possible d'obtenir une sélection de ressources en spécifiant la syntaxe suivante :

```
http://<hote>:<port>/manager/resources?type=<type_JNDI>
```

serverinfo

Saisissez la commande suivante :

```
[root@centos7 bin]# lynx --dump -auth admin:fenestros "http://www.i2tch.loc:8080/manager/text/serverinfo"
OK - Server info
Tomcat Version: Apache Tomcat/8.0.36
OS Name: Linux
OS Version: 3.10.0-1062.4.1.el7.x86_64
OS Architecture: amd64
JVM Version: 1.8.0_232-b09
JVM Vendor: Oracle Corporation
```

L'interface HTML

Afin de pouvoir utiliser l'application manager en mode html, modifiez le fichier **$CATALINA_HOME/conf/tomcat-users.xml** ainsi :

```
<?xml version='1.0' encoding='utf-8'?>
<tomcat-users xmlns="http://tomcat.apache.org/xml"
              xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
              xsi:schemaLocation="http://tomcat.apache.org/xml tomcat-users.xsd"
              version="1.0">
  <role rolename="tomcat"/>
  <role rolename="role1"/>
  <role rolename="manager-gui"/>
```

```
  <user username="tomcat" password="tomcat" roles="tomcat"/>
  <user username="both" password="tomcat" roles="tomcat,role1"/>
  <user username="role1" password="tomcat" roles="role1"/>
  <user username="admin" password="fenestros" roles="manager-gui"/>
</tomcat-users>
```

Arrêtez puis démarrez le serveur Tomcat :

```
[root@centos7 bin]# ./shutdown.sh
Using CATALINA_BASE:   /usr/tomcat8
Using CATALINA_HOME:   /usr/tomcat8
Using CATALINA_TMPDIR: /usr/tomcat8/temp
Using JRE_HOME:        /usr/lib/jvm/jre-1.8.0-openjdk-1.8.0.232.b09-0.el7_7.x86_64
Using CLASSPATH:       /usr/tomcat8/bin/bootstrap.jar:/usr/tomcat8/bin/tomcat-juli.jar
[root@centos7 bin]# ./startup.sh
Using CATALINA_BASE:   /usr/tomcat8
Using CATALINA_HOME:   /usr/tomcat8
Using CATALINA_TMPDIR: /usr/tomcat8/temp
Using JRE_HOME:        /usr/lib/jvm/jre-1.8.0-openjdk-1.8.0.232.b09-0.el7_7.x86_64
Using CLASSPATH:       /usr/tomcat8/bin/bootstrap.jar:/usr/tomcat8/bin/tomcat-juli.jar
Tomcat started.
```

Si vous vous êtes connecté à votre machine virtuelle via ssh, passez la VM CentOS 7 en démarrage en mode graphique :

```
[root@centos7 bin]# ls -l /etc/systemd/system/default.target
lrwxrwxrwx. 1 root root 37 Apr 30  2016 /etc/systemd/system/default.target -> /lib/systemd/system/multi-
user.target
[root@centos7 bin]# rm -rf /etc/systemd/system/default.target
[root@centos7 bin]# ln -s /lib/systemd/system/graphical.target /etc/systemd/system/default.target
[root@centos7 bin]# ls -l /etc/systemd/system/default.target
lrwxrwxrwx. 1 root root 36 Oct 29 13:27 /etc/systemd/system/default.target ->
/lib/systemd/system/graphical.target
[root@centos7 bin]# shutdown -h now
```

Une fois la VM arrêtée, augmentez la mémoire qui lui est allouée à au moins 2Go (plus si votre machine hôte le permet), puis démarrez votre VM.

Connectez-vous à votre machine virtuelle et **démarrez** Tomcat via ssh et démarrez Tomcat :

```
[root@centos7 bin]# ./startup.sh
Using CATALINA_BASE:   /usr/tomcat8
Using CATALINA_HOME:   /usr/tomcat8
Using CATALINA_TMPDIR: /usr/tomcat8/temp
Using JRE_HOME:        /usr/lib/jvm/jre-1.8.0-openjdk-1.8.0.232.b09-0.el7_7.x86_64
Using CLASSPATH:       /usr/tomcat8/bin/bootstrap.jar:/usr/tomcat8/bin/tomcat-juli.jar
Tomcat started.
```

Lancez ensuite le navigateur web Firefox dans la fenêtre de la VM et saisissez l'url http://www.i2tch.loc:8080/manager/html.

Dans la boîte d'authentification renseignez l'utilisateur **admin** et le mot de passe **fenestros**.

Validez. Vous obtiendrez l'interface web de gestion de Tomcat.

Explorez cette interface. Ensuite passez en revue chacun des exemples dans les applications **jsp-examples** et **servlets-examples** en affichant le résultat ainsi que le code.

L'interface ANT

ANT est un utilitaire permettant la gestion des applications par l'utilisation d'un script normalement appelé **build.xml** contenant un **projet**.

Commencez par l'installation d'ANT :

```
[root@centos7 bin]# yum install ant
```

Définissez la variable CLASSPATH dans le fichier /etc/profile :

```
[root@centos7 bin]# vi /etc/profile
[root@centos7 bin]# cat /etc/profile
```

```
...
unset -f pathmunge
# Tomcat
CATALINA_HOME="/usr/tomcat8"
export CATALINA_HOME
PATH=$PATH:/usr/lib/jvm/jre-1.8.0-openjdk-1.8.0.232.b09-0.el7_7.x86_64/bin
JAVA_HOME=/usr/lib/jvm/jre-1.8.0-openjdk-1.8.0.232.b09-0.el7_7.x86_64
export PATH JAVA_HOME
# CLASSPATH
CLASSPATH="/usr/tomcat8/lib:/usr/share/java:/usr/share/ant/lib"
[root@centos7 bin]#
[root@centos7 bin]# source /etc/profile
[root@centos7 bin]#
[root@centos7 bin]# echo $CLASSPATH
/usr/tomcat8/lib:/usr/share/java:/usr/share/ant/lib
```

Donnez le rôle **manager-script** à l'utilisateur **admin** :

```
[root@centos7 bin]# vi $CATALINA_HOME/conf/tomcat-users.xml
[root@centos7 bin]# cat $CATALINA_HOME/conf/tomcat-users.xml
<?xml version='1.0' encoding='utf-8'?>
<tomcat-users xmlns="http://tomcat.apache.org/xml"
              xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
              xsi:schemaLocation="http://tomcat.apache.org/xml tomcat-users.xsd"
              version="1.0">
  <role rolename="tomcat"/>
  <role rolename="role1"/>
  <role rolename="manager-script"/>
  <user username="tomcat" password="tomcat" roles="tomcat"/>
  <user username="both" password="tomcat" roles="tomcat,role1"/>
  <user username="role1" password="tomcat" roles="role1"/>
  <user username="admin" password="fenestros" roles="manager-script"/>
</tomcat-users>
```

Re-démarrez le serveur Tomcat pour une prise en compte de la modification :

```
[root@centos7 bin]# ./shutdown.sh
Using CATALINA_BASE:   /usr/tomcat8
Using CATALINA_HOME:   /usr/tomcat8
Using CATALINA_TMPDIR: /usr/tomcat8/temp
Using JRE_HOME:        /usr/lib/jvm/jre-1.8.0-openjdk-1.8.0.232.b09-0.el7_7.x86_64
Using CLASSPATH:       /usr/tomcat8/bin/bootstrap.jar:/usr/tomcat8/bin/tomcat-juli.jar
[root@centos7 bin]# ./startup.sh
Using CATALINA_BASE:   /usr/tomcat8
Using CATALINA_HOME:   /usr/tomcat8
Using CATALINA_TMPDIR: /usr/tomcat8/temp
Using JRE_HOME:        /usr/lib/jvm/jre-1.8.0-openjdk-1.8.0.232.b09-0.el7_7.x86_64
Using CLASSPATH:       /usr/tomcat8/bin/bootstrap.jar:/usr/tomcat8/bin/tomcat-juli.jar
Tomcat started.
```

Créez le script **build.xml** dans le répertoire courant pour déployer l'application **sample** :

```
[root@centos7 bin]# vi build.xml
[root@centos7 bin]# cat build.xml
<?xml version="1.0" encoding="ISO-8859-1" ?>
<project name="Tomcat" default="deployer" basedir=".">
<property name="manager.url" value="http://www.i2tch.loc:8080/manager/text" />
<property name="manager.user" value="admin" />
<property name="manager.password" value="fenestros" />
<property name="app.context" value="/sample" />
<property name="app.war" value="file:/usr/tomcat8/webapps/docs/appdev/sample/sample.war" />
<property name="appserver.home" value="/usr/tomcat8" />
<property name="appserver.lib" value="${appserver.home}/lib" />
<path id="catalina-ant-classpath">
   <fileset dir="${appserver.lib}">
      <include name="*.jar"/>
   </fileset>
</path>
```

```
<taskdef name="deploy" classname="org.apache.catalina.ant.DeployTask">
    <classpath refid="catalina-ant-classpath"/>
</taskdef>
<taskdef name="reload" classname="org.apache.catalina.ant.ReloadTask">
    <classpath refid="catalina-ant-classpath"/>
</taskdef>
<taskdef name="undeploy" classname="org.apache.catalina.ant.UndeployTask">
    <classpath refid="catalina-ant-classpath"/>
</taskdef>
<taskdef name="list" classname="org.apache.catalina.ant.ListTask">
    <classpath refid="catalina-ant-classpath"/>
</taskdef>
<taskdef name="start" classname="org.apache.catalina.ant.StartTask">
    <classpath refid="catalina-ant-classpath"/>
</taskdef>
<taskdef name="stop" classname="org.apache.catalina.ant.StopTask">
    <classpath refid="catalina-ant-classpath"/>
</taskdef>
<target name="deployer" description="Déploiement">
        <deploy url="${manager.url}" username="${manager.user}" password="${manager.password}"
path="${app.context}" war="${app.war}" />
</target>
<target name="recharger" description="Rechargement">
        <reload url="${manager.url}" username="${manager.user}" password="${manager.password}"
path="${app.context}" />
</target>
<target name="supprimer" description="Suppression">
        <undeploy url="${manager.url}" username="${manager.user}" password="${manager.password}"
path="${app.context}" />
</target>
<target name="demarrer" description="Démarrage">
        <start url="${manager.url}" username="${manager.user}" password="${manager.password}"
path="${app.context}" />
</target>
```

```
<target name="arreter" description="Arrêt">
        <stop url="${manager.url}" username="${manager.user}" password="${manager.password}"
path="${app.context}" />
</target>
</project>
```

Pour tester le fichier **/usr/tomcat8/bin/build.xml**, vérifiez que vous pouvez déployer l'application **sample** avec la commande **ant deployer** :

```
[root@centos7 bin]# ant deployer
Buildfile: /usr/tomcat8/bin/build.xml

deployer:
   [deploy] OK - Deployed application at context path /sample

BUILD SUCCESSFUL
Total time: 3 seconds
```

Deployer de Tomcat

L'outil **Deployer** n'est pas inclus avec les binaires de Tomcat. Pour cette raison il convient de le télécharger :

```
[root@centos7 ~]# wget
https://archive.apache.org/dist/tomcat/tomcat-8/v8.0.36/bin/apache-tomcat-8.0.36-deployer.tar.gz
```

Désarchivez le fichier apache-tomcat-8.0.36-deployer.tar.gz :

```
[root@centos7 ~]# tar xvf apache-tomcat-8.0.36-deployer.tar.gz
apache-tomcat-8.0.36-deployer/LICENSE
apache-tomcat-8.0.36-deployer/NOTICE
apache-tomcat-8.0.36-deployer/RELEASE-NOTES
apache-tomcat-8.0.36-deployer/images/
apache-tomcat-8.0.36-deployer/lib/
```

```
apache-tomcat-8.0.36-deployer/build.xml
apache-tomcat-8.0.36-deployer/deployer-howto.html
apache-tomcat-8.0.36-deployer/images/asf-logo.gif
apache-tomcat-8.0.36-deployer/images/tomcat.gif
apache-tomcat-8.0.36-deployer/lib/catalina-ant.jar
apache-tomcat-8.0.36-deployer/lib/catalina-deployer.jar
apache-tomcat-8.0.36-deployer/lib/el-api.jar
apache-tomcat-8.0.36-deployer/lib/jasper-el.jar
apache-tomcat-8.0.36-deployer/lib/jasper.jar
apache-tomcat-8.0.36-deployer/lib/jsp-api.jar
apache-tomcat-8.0.36-deployer/lib/servlet-api.jar
apache-tomcat-8.0.36-deployer/lib/tomcat-coyote.jar
apache-tomcat-8.0.36-deployer/lib/tomcat-juli.jar
apache-tomcat-8.0.36-deployer/lib/tomcat-util-scan.jar
apache-tomcat-8.0.36-deployer/lib/tomcat-util.jar
apache-tomcat-8.0.36-deployer/lib/tomcat-websocket.jar
apache-tomcat-8.0.36-deployer/lib/websocket-api.jar
```

Cette application contient un fichier build.xml déjà partiellement configuré :

```
[root@centos7 ~]# cat apache-tomcat-8.0.36-deployer/build.xml
<?xml version='1.0' encoding='utf-8'?>
<!--
  Licensed to the Apache Software Foundation (ASF) under one or more
  contributor license agreements.  See the NOTICE file distributed with
  this work for additional information regarding copyright ownership.
  The ASF licenses this file to You under the Apache License, Version 2.0
  (the "License"); you may not use this file except in compliance with
  the License.  You may obtain a copy of the License at

      http://www.apache.org/licenses/LICENSE-2.0

  Unless required by applicable law or agreed to in writing, software
  distributed under the License is distributed on an "AS IS" BASIS,
```

```
  WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
  See the License for the specific language governing permissions and
  limitations under the License.
-->
<project name="Deployer" default="compile" basedir=".">

  <property file="deployer.properties"/>

  <!-- Configure the directory into which the web application is built -->
  <property name="build"    value="${basedir}/build"/>

  <!-- Configure the folder and context path for this application -->
  <property name="webapp"   value="myapp"/>
  <property name="path"     value="/myapp"/>

  <!-- Configure properties to access the Manager application -->
  <property name="url"      value="http://localhost:8080/manager/text"/>
  <property name="username" value="tomcat"/>
  <property name="password" value="tomcat"/>

  <property name="webapp.path"     value="${build}/webapp${path}"/>

  <path id="deployer.classpath">
    <fileset dir="${basedir}/lib">
      <include name="*.jar"/>
    </fileset>
  </path>

  <!-- Configure the custom Ant tasks for the Manager application -->
  <taskdef resource="org/apache/catalina/ant/catalina.tasks"
           classpathref="deployer.classpath"/>

  <!-- Executable Targets -->
  <target name="clean" description="Removes build directory">
```

```
    <delete dir="${build}" />
  </target>

  <target name="compile" description="Compile web application"
          depends="clean">

    <copy todir="${webapp.path}">
      <fileset dir="${webapp}" />
    </copy>

    <jasper validateXml="false"
            uriroot="${webapp.path}"
            webXmlFragment="${webapp.path}/WEB-INF/generated_web.xml"
            addWebXmlMappings="true"
            outputDir="${webapp.path}/WEB-INF/classes" />

    <validator path="${webapp.path}" />

    <mkdir dir="${webapp.path}/WEB-INF/classes"/>
    <mkdir dir="${webapp.path}/WEB-INF/lib"/>

    <javac destdir="${webapp.path}/WEB-INF/classes"
           optimize="off"
           debug="${compile.debug}"
           deprecation="${compile.deprecation}"
           failonerror="false"
           srcdir="${webapp.path}/WEB-INF/classes"
           encoding="UTF-8"
           excludes="**/*.smap">
      <classpath>
        <fileset dir="${webapp.path}/WEB-INF/lib">
          <include name="*.jar"/>
        </fileset>
        <fileset dir="${basedir}/lib">
```

```
        <include name="*.jar"/>
      </fileset>
    </classpath>
    <include name="**" />
    <exclude name="tags/**" />
  </javac>

  <jar destfile="${webapp.path}.war"
       basedir="${webapp.path}" />

</target>

<target name="deploy" description="Deploy web application">
  <deploy url="${url}" username="${username}" password="${password}"
          path="${path}" war="${webapp.path}.war" update="true" />
</target>

<target name="undeploy" description="Undeploy web application">
  <undeploy url="${url}" username="${username}" password="${password}"
            path="${path}"/>
</target>

<!-- Webapp lifecycle control -->
<target name="start" description="Start web application">
  <start url="${url}" username="${username}" password="${password}"
         path="${path}"/>
</target>
<target name="reload" description="Reload web application">
  <reload url="${url}" username="${username}" password="${password}"
          path="${path}"/>
</target>
<target name="stop" description="Stop web application">
  <stop url="${url}" username="${username}" password="${password}"
        path="${path}"/>
```

```
  </target>

</project>
```

Pour personnaliser ce script, il faut créer le fichier **deployer.properties** au même endroit que le script lui-même. Prenez 30 minutes pour chercher sur Internet comment configurer et utiliser le Deployer.

SER305 - Sécurité du serveur Tomcat 8

Contenu du Module

- Tomcat et le SSL
  - Présentation de SSL
    - Fonctionnement de SSL
  - Configurer Tomcat
  - Configurer Apache
    - Installation de SSL
    - Configuration de SSL
    - Mise en place des paramètres de sécurité SSL
    - Tester Votre Configuration
  - Apache en Frontal HTTPS
  - Restrictions d'Accès
  - Le Gestionnaire de Sécurité

Authentification, Autorisation et Cryptage

Authentification

Quand un client tente d'accéder à une ressource protégée d'un site ou d'une application web, le serveur renvoie un code HTTP **401**. A la réception de ce code, la navigateur affiche une boîte de dialogue d'authentification. Dans le cas où l'authentification n'aboutit pas, le serveur envoie un code HTTP **403** (Forbidden).

Pour mettre en œuvre ce mécanisme il existe quatre schémas d'authentification, dont les trois premiers sont :

- **BASIC** - l'utilisation de l'algorithme **Base64**,
- **DIGEST**, - l'utilisation d'un algorithme de hachage tel **SHA** ou **MD5**,
- **CLIENT-CERT** - l'utilisation de certificats HTTPS.

Les deux derniers schémas ci-dessus ne sont pas supportés par tous les navigateurs. Par conséquent, l'utilisation de l'authentification de base et HTTPS ensemble est plus courant.

Le quatrième schéma d'authentification est l'authentification par formulaire, **FORM**. Dans ce cas, le navigateur n'intervient pas car c'est le serveur qui sert un formulaire HTML au client.

Autorisation

Tomcat utilise un système **RBAC** (*Role Based Access Control*) pour l'autorisation. Dans ce cas, chaque utilisateur est attribué un ou plusieurs rôles dans le **registre d'authentification**.

Cryptage

Tomcat peut utiliser soit **SSL** soit **TLS** pour sécuriser le flux de données HTTP. Les technologies Java utilise les protocoles SSL et TLS dans la bibliothèque **JSSE** (*Java Secure Socket Extention*).

La Sécurité sous Tomcat

Configuration

La configuration de la sécurité se fait dans le fichier **web.xml** de l'application concernée. Consultez le fichier **/usr/tomcat8/webapps/examples/WEB-INF/web.xml**. A la fin de celui-ci, trouvez l'élément **<security-constraint>** :

```
...
   <security-constraint>
      <display-name>Example Security Constraint - part 1</display-name>
      <web-resource-collection>
         <web-resource-name>Protected Area - Allow methods</web-resource-name>
         <!-- Define the context-relative URL(s) to be protected -->
         <url-pattern>/jsp/security/protected/*</url-pattern>
         <!-- If you list http methods, only those methods are protected so -->
         <!-- the constraint below ensures all other methods are denied     -->
         <http-method>DELETE</http-method>
         <http-method>GET</http-method>
         <http-method>POST</http-method>
```

```
        <http-method>PUT</http-method>
      </web-resource-collection>
      <auth-constraint>
        <!-- Anyone with one of the listed roles may access this area -->
        <role-name>tomcat</role-name>
        <role-name>role1</role-name>
      </auth-constraint>
    </security-constraint>
  <security-constraint>
      <display-name>Example Security Constraint - part 2</display-name>
      <web-resource-collection>
        <web-resource-name>Protected Area - Deny methods</web-resource-name>
        <!-- Define the context-relative URL(s) to be protected -->
        <url-pattern>/jsp/security/protected/*</url-pattern>
        <http-method-omission>DELETE</http-method-omission>
        <http-method-omission>GET</http-method-omission>
        <http-method-omission>POST</http-method-omission>
        <http-method-omission>PUT</http-method-omission>
      </web-resource-collection>
      <!-- An empty auth constraint denies access -->
      <auth-constraint />
    </security-constraint>

    <!-- Default login configuration uses form-based authentication -->
    <login-config>
      <auth-method>FORM</auth-method>
      <realm-name>Example Form-Based Authentication Area</realm-name>
      <form-login-config>
        <form-login-page>/jsp/security/protected/login.jsp</form-login-page>
        <form-error-page>/jsp/security/protected/error.jsp</form-error-page>
      </form-login-config>
    </login-config>

    <!-- Security roles referenced by this web application -->
```

```
    <security-role>
      <role-name>role1</role-name>
    </security-role>
    <security-role>
      <role-name>tomcat</role-name>
    </security-role>
...
```

L'élément <security-constraint> contient d'autres éléments dont les plus importants sont :

Elément	**Description**
<web-resource-collection></web-resource-collection>	Contient les ressources à protéger
<auth-constraint></auth-constraint>	Indique les rôles qui auront accès au ressources protégées

L'élément <Login-conf> contient d'autres éléments dont les plus importants sont :

Elément	**Description**
<auth-method></auth-method>	Vaut BASIC, DIGEST, CLIENT-CERT ou FORM
<form-login-page></form-login-page>	Indique la page contenant le formulaire
<form-error-page></form-error-page>	Indique la page d'erreur envoyée au client en cas d'échec d'authentification

L'élément <Security-role> doit contenir un élément **<role-name>** pour chaque rôle à déclarer.

Pour utiliser l'authentification par formulaire, celui-ci doit :

- être posté à destination du servlet **j_security_check**,
- posséder le champ **j_username** pour recevoir le nom de l'utilisateur,
- posséder le champ **j_password** pour recevoir le mote de passe de l'utilisateur.

Realms

L'accès au **registre d'authentification** est obtenu en utilisant des **Realms**. Tomcat peut utiliser les sept Realms suivants :

- User Database Realm,
- JDBC Realm,
- DataSource Realm,
- JNDI Realm,
- JAAS Realm,
- Combined Realm,
- LockOut Realm.

User Database Realm

Ce type de Realm utilise la classe Java **org.apache.catalina.realm.UserDatabaseRealm**. Les informations sont stockées dans un fichier XML qui est par défaut **$CATALINA_HOME/conf/tomcat-users.xml** :

```
<?xml version='1.0' encoding='utf-8'?>
<tomcat-users xmlns="http://tomcat.apache.org/xml"
              xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
              xsi:schemaLocation="http://tomcat.apache.org/xml tomcat-users.xsd"
              version="1.0">
  <role rolename="tomcat"/>
  <role rolename="role1"/>
  <role rolename="manager-script"/>
  <user username="tomcat" password="tomcat" roles="tomcat"/>
  <user username="both" password="tomcat" roles="tomcat,role1"/>
  <user username="role1" password="tomcat" roles="role1"/>
  <user username="admin" password="fenestros" roles="manager-script"/>
</tomcat-users>
```

La configuration de ce Realm se trouve dans le fichier **$CATALINA_HOME/conf/server.xml** dans les éléments <GlobalNamingResources> et <Engine> :

```
  <GlobalNamingResources>
    <!-- Editable user database that can also be used by
         UserDatabaseRealm to authenticate users
```

```
    -->
    <Resource name="UserDatabase" auth="Container"
              type="org.apache.catalina.UserDatabase"
              description="User database that can be updated and saved"
              factory="org.apache.catalina.users.MemoryUserDatabaseFactory"
              pathname="conf/tomcat-users.xml" />
  </GlobalNamingResources>
  ...
    <Engine name="Catalina" defaultHost="localhost">
  ...
        <Realm className="org.apache.catalina.realm.UserDatabaseRealm"
               resourceName="UserDatabase"/>
      </Realm>
  ...
```

Dans le cas ci-dessus, les mots de passe sont en clair dans le fichier $CATALINA_HOME/conf/tomcat-users.xml. Il est cependant possible de les cryptés grâce à la classe **javax.security.MessageDigest** en utilisant soit l'algorithme **SHA** soit l'algorithme **MD5** :

```
[root@centos7 ~]# cd $CATALINA_HOME/bin
[root@centos7 bin]# ./digest.sh -a sha fenestros
fenestros:75affcc6adf7ec7cd21f6479b8fb87b89a550b2602e613715d57d862b15c7c17$1$015b4b320315c87572918dbcc08b65a240d0
a750
```

Il est ensuite nécessaire d'éditer le fichier **$CATALINA_HOME/conf/tomcat-users.xml** en remplacant le mot de passe en clair “fenestros” avec le mot de passe crypté :

```
[root@centos7 bin]# vi $CATALINA_HOME/conf/tomcat-users.xml
[root@centos7 bin]# cat $CATALINA_HOME/conf/tomcat-users.xml
<?xml version='1.0' encoding='utf-8'?>
<tomcat-users xmlns="http://tomcat.apache.org/xml"
              xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
              xsi:schemaLocation="http://tomcat.apache.org/xml tomcat-users.xsd"
              version="1.0">
  <role rolename="tomcat"/>
```

```
  <role rolename="role1"/>
  <role rolename="manager-script"/>
  <user username="tomcat" password="tomcat" roles="tomcat"/>
  <user username="both" password="tomcat" roles="tomcat,role1"/>
  <user username="role1" password="tomcat" roles="role1"/>
  <user username="admin"
password="75affcc6adf7ec7cd21f6479b8fb87b89a550b2602e613715d57d862b15c7c17$1$015b4b320315c87572918dbcc08b65a240d0
a750" roles="manager-script"/>
</tomcat-users>
```

> **Important** : NE COPIEZ PAS simplement l'exemple du fichier ci-dessus. MODIFIEZ le fichier en remplaçant le mot de passe fenestros avec le mot de passe crypté que VOUS obtenez en exécutant la commande **./digest.sh -a sha fenestros**.

Dernièrement il faut éditer le fichier **$CATALINA_HOME/conf/server.xml**

```
...
    <Engine name="Catalina" defaultHost="localhost">
  ...
        <Realm className="org.apache.catalina.realm.UserDatabaseRealm"
               resourceName="UserDatabase" digest="sha" />
      </Realm>
...
```

Redémarrez le serveur Tomcat :

```
[root@centos7 bin]# ./shutdown.sh
Using CATALINA_BASE:   /usr/tomcat8
Using CATALINA_HOME:   /usr/tomcat8
Using CATALINA_TMPDIR: /usr/tomcat8/temp
Using JRE_HOME:        /usr/lib/jvm/jre-1.8.0-openjdk-1.8.0.232.b09-0.el7_7.x86_64
Using CLASSPATH:       /usr/tomcat8/bin/bootstrap.jar:/usr/tomcat8/bin/tomcat-juli.jar
```

```
[root@centos7 bin]# ./startup.sh
Using CATALINA_BASE:   /usr/tomcat8
Using CATALINA_HOME:   /usr/tomcat8
Using CATALINA_TMPDIR: /usr/tomcat8/temp
Using JRE_HOME:        /usr/lib/jvm/jre-1.8.0-openjdk-1.8.0.232.b09-0.el7_7.x86_64
Using CLASSPATH:       /usr/tomcat8/bin/bootstrap.jar:/usr/tomcat8/bin/tomcat-juli.jar
Tomcat started.
```

Testez votre connexion :

```
[root@centos7 bin]# lynx --dump -auth admin:fenestros "http://www.i2tch.loc:8080/manager/text/serverinfo"
OK - Server info
Tomcat Version: Apache Tomcat/8.0.36
OS Name: Linux
OS Version: 3.10.0-1062.4.1.el7.x86_64
OS Architecture: amd64
JVM Version: 1.8.0_232-b09
JVM Vendor: Oracle Corporation
```

JDBC Realm

Ce type de Realm utilise la classe Java **org.apache.catalina.realm.JDBCRealm**. Les informations sont stockées dans une base de données dont le script MySQL est le suivant :

```
CREATE DATABASE `auth_tomcat`;
USE `auth_tomcat`;
CREATE TABLE `auth_tomcat`.`users` (
        `nom_user` varchar(45) NOT NULL,
        `mdp_user` varchar(45) NOT NULL,
        PRIMARY KEY (`nom_user`)
) ENGINE=InnoDB DEFAULT CHARSET=latin1;
CREATE TABLE `auth_tomcat`.`roles` (
```

```
        `nom_user` varchar(45) NOT NULL,
        `nom_role` varchar(45) NOT NULL,
        PRIMARY KEY (`nom_user`)
) ENGINE=InnoDB DEFAULT CHARSET=latin1;
```

Par exemple :

```
[root@centos7 bin]# cd ~
[root@centos7 ~]# vi auth_tomcat
[root@centos7 ~]# cat auth_tomcat
CREATE DATABASE `auth_tomcat`;
USE `auth_tomcat`;
CREATE TABLE `auth_tomcat`.`users` (
        `nom_user` varchar(45) NOT NULL,
        `mdp_user` varchar(45) NOT NULL,
        PRIMARY KEY (`nom_user`)
) ENGINE=InnoDB DEFAULT CHARSET=latin1;
CREATE TABLE `auth_tomcat`.`roles` (
        `nom_user` varchar(45) NOT NULL,
        `nom_role` varchar(45) NOT NULL,
        PRIMARY KEY (`nom_user`)
) ENGINE=InnoDB DEFAULT CHARSET=latin1;
```

Créez donc la basee de données **auth_tomcat** :

```
[root@centos7 ~]# mysql -u root -p < auth_tomcat
Enter password: fenestros
[root@centos7 ~]#
```

Connectez-vous au serveur MariaDB et créez l'utilisateur **admin1** ayant un mot de passe **fenestros** et un rôle **manager-script** :

```
[root@centos7 ~]# mysql -u root -p
Enter password: fenestros
Welcome to the MariaDB monitor.  Commands end with ; or \g.
```

```
Your MariaDB connection id is 12
Server version: 5.5.47-MariaDB MariaDB Server

Copyright (c) 2000, 2015, Oracle, MariaDB Corporation Ab and others.

Type 'help;' or '\h' for help. Type '\c' to clear the current input statement.

MariaDB [(none)]> SHOW databases;
+--------------------+
| Database           |
+--------------------+
| information_schema |
| auth_tomcat        |
| mysql              |
| performance_schema |
| test               |
| tomcat             |
+--------------------+
6 rows in set (0.01 sec)

MariaDB [(none)]> USE auth_tomcat;
Reading table information for completion of table and column names
You can turn off this feature to get a quicker startup with -A

Database changed
MariaDB [auth_tomcat]> INSERT INTO `auth_tomcat`.`users` VALUES('admin1','fenestros');
Query OK, 1 row affected (0.01 sec)

MariaDB [auth_tomcat]> INSERT INTO `auth_tomcat`.`roles` VALUES('admin1','manager-script');
Query OK, 1 row affected (0.02 sec)

MariaDB [auth_tomcat]> GRANT SELECT ON auth_tomcat.* TO 'tomcat'@'localhost' IDENTIFIED BY 'tomcat';
Query OK, 0 rows affected (0.10 sec)
```

```
MariaDB [auth_tomcat]> FLUSH PRIVILEGES;
Query OK, 0 rows affected (0.02 sec)

MariaDB [auth_tomcat]> SET PASSWORD FOR 'tomcat'@'localhost' = PASSWORD('secret');
Query OK, 0 rows affected (0.02 sec)

MariaDB [auth_tomcat]> exit
Bye
```

Modifiez le fichier **$CATALINA_HOME/conf/server.xml** en mettant en commentaires le Realm **In-Memory** et en ajoutant la section suivante :

```
...
        <!-- <Realm className="org.apache.catalina.realm.UserDatabaseRealm"
               resourceName="UserDatabase" digest="sha" /> -->

      <Realm  className="org.apache.catalina.realm.JDBCRealm"
             driverName="com.mysql.jdbc.Driver"
          connectionURL="jdbc:mysql://localhost:3306/auth_tomcat"
         connectionName="tomcat" connectionPassword="secret"
              userTable="users" userNameCol="nom_user" userCredCol="mdp_user"
          userRoleTable="roles" roleNameCol="nom_role" />

      </Realm>
...
```

Redémarrez le serveur Tomcat :

```
[root@centos7 ~]# cd $CATALINA_HOME/bin
[root@centos7 bin]# ./shutdown.sh
Using CATALINA_BASE:   /usr/tomcat8
Using CATALINA_HOME:   /usr/tomcat8
Using CATALINA_TMPDIR: /usr/tomcat8/temp
Using JRE_HOME:        /usr/lib/jvm/jre-1.8.0-openjdk-1.8.0.232.b09-0.el7_7.x86_64
Using CLASSPATH:       /usr/tomcat8/bin/bootstrap.jar:/usr/tomcat8/bin/tomcat-juli.jar
```

```
[root@centos7 bin]# ./startup.sh
Using CATALINA_BASE:   /usr/tomcat8
Using CATALINA_HOME:   /usr/tomcat8
Using CATALINA_TMPDIR: /usr/tomcat8/temp
Using JRE_HOME:        /usr/lib/jvm/jre-1.8.0-openjdk-1.8.0.232.b09-0.el7_7.x86_64
Using CLASSPATH:       /usr/tomcat8/bin/bootstrap.jar:/usr/tomcat8/bin/tomcat-juli.jar
Tomcat started.
```

Vérifiez maintenant que vous pouvez vous connecter avec le compte **admin1** :

```
[root@centos7 bin]# lynx --dump -auth admin1:fenestros "http://www.i2tch.loc:8080/manager/text/serverinfo"
OK - Server info
Tomcat Version: Apache Tomcat/8.0.36
OS Name: Linux
OS Version: 3.10.0-1062.4.1.el7.x86_64
OS Architecture: amd64
JVM Version: 1.8.0_232-b09
JVM Vendor: Oracle Corporation
```

DataSource Realm

Ce type de Realm utilise la classe Java **org.apache.catalina.realm.DataSourceRealm**. Les informations sont stockées dans une base de données identique au cas du Realm JDBCRealm. La différence principale du Realm DataSource par rapport au Realm JDBC est que le premier peut utiliser un **pool** de connexions augmentant ainsi la performance.

Créez d'abord l'utilisateur **admin2** dans la base de données **auth_tomcat** :

```
[root@centos7 bin]# mysql -u root -p
Enter password:
Welcome to the MariaDB monitor.  Commands end with ; or \g.
Your MariaDB connection id is 2
Server version: 5.5.47-MariaDB MariaDB Server
```

```
Copyright (c) 2000, 2015, Oracle, MariaDB Corporation Ab and others.

Type 'help;' or '\h' for help. Type '\c' to clear the current input statement.

MariaDB [(none)]> USE auth_tomcat;
Reading table information for completion of table and column names
You can turn off this feature to get a quicker startup with -A

Database changed
MariaDB [auth_tomcat]> INSERT INTO `auth_tomcat`.`users` VALUES('admin2','fenestros');
Query OK, 1 row affected (0.01 sec)

MariaDB [auth_tomcat]> INSERT INTO `auth_tomcat`.`roles` VALUES('admin2','manager-script');
Query OK, 1 row affected (0.00 sec)

MariaDB [auth_tomcat]> exit
Bye
[root@centos7 bin]#
```

Modifiez ensuite le fichier **$CATALINA_HOME/conf/server.xml** en y ajoutant un élément **<Resource name="jdbc/AuthTomcat" ...>** dans l'élément **<GlobalNamingResource>** :

```
...
  <GlobalNamingResources>
    <!-- Editable user database that can also be used by
         UserDatabaseRealm to authenticate users
    -->
    <Resource name="UserDatabase" auth="Container"
              type="org.apache.catalina.UserDatabase"
              description="User database that can be updated and saved"
              factory="org.apache.catalina.users.MemoryUserDatabaseFactory"
              pathname="conf/tomcat-users.xml" />

    <Resource name="jdbc/AuthTomcat" auth="Container"
```

```
              type="javax.sql.DataSource"
       driverName="com.mysql.jdbc.Driver"
              url="jdbc:mysql://localhost:3306/auth_tomcat"
         username="tomcat"
         password="secret" />

  </GlobalNamingResources>
...
```

Ensuite, éditez l'élément Realm précédemment créé ainsi :

```
...
       <!-- <Realm className="org.apache.catalina.realm.UserDatabaseRealm"
              resourceName="UserDatabase" digest="sha" /> -->

     <!-- <Realm  className="org.apache.catalina.realm.JDBCRealm"
            driverName="com.mysql.jdbc.Driver"
         connectionURL="jdbc:mysql://localhost:3306/auth_tomcat"
        connectionName="tomcat" connectionPassword="secret"
             userTable="users" userNameCol="nom_user" userCredCol="mdp_user"
         userRoleTable="roles" roleNameCol="nom_role" />

     </Realm> -->

     <Realm  className="org.apache.catalina.realm.DataSourceRealm"
        dataSourceName="jdbc/AuthTomcat"
             userTable="users" userNameCol="nom_user" userCredCol="mdp_user"
         userRoleTable="roles" roleNameCol="nom_role" />
    </Realm>
...
```

Redémarrez le serveur Tomcat :

```
[root@centos7 bin]# ./shutdown.sh
```

```
Using CATALINA_BASE:   /usr/tomcat8
Using CATALINA_HOME:   /usr/tomcat8
Using CATALINA_TMPDIR: /usr/tomcat8/temp
Using JRE_HOME:        /usr/lib/jvm/jre-1.8.0-openjdk-1.8.0.232.b09-0.el7_7.x86_64
Using CLASSPATH:       /usr/tomcat8/bin/bootstrap.jar:/usr/tomcat8/bin/tomcat-juli.jar
[root@centos7 bin]# ./startup.sh
Using CATALINA_BASE:   /usr/tomcat8
Using CATALINA_HOME:   /usr/tomcat8
Using CATALINA_TMPDIR: /usr/tomcat8/temp
Using JRE_HOME:        /usr/lib/jvm/jre-1.8.0-openjdk-1.8.0.232.b09-0.el7_7.x86_64
Using CLASSPATH:       /usr/tomcat8/bin/bootstrap.jar:/usr/tomcat8/bin/tomcat-juli.jar
Tomcat started.
```

Vérifiez maintenant que vous pouvez vous connecter avec le compte **admin2** :

```
[root@centos7 bin]# lynx --dump -auth admin2:fenestros "http://www.i2tch.loc:8080/manager/text/serverinfo"
OK - Server info
Tomcat Version: Apache Tomcat/8.0.36
OS Name: Linux
OS Version: 3.10.0-1062.4.1.el7.x86_64
OS Architecture: amd64
JVM Version: 1.8.0_232-b09
JVM Vendor: Oracle Corporation
```

JNDI Realm

Ce type de Realm utilise la classe Java **org.apache.catalina.realm.JNDIRealm**. Les informations sont stockées dans une base de données LDAP.

Le format LDIF

Les fichiers au format LDIF (**LD**AP **I**nterchange **F**ormat) sont utilisés lors de modifications de masse sur une base LDAP. Les fichiers LDIF sont traités

dans un ordre séquentielle.

Le fichier LDIF est un fichier texte qui peut comprendre :

- des descriptions d'entrées de l'annuaire,
- des valeurs d'attribut pour les entrées de l'annuaire,
- des instructions de traitements pour le serveur.

Un fichier LDIF peut comporter des commentaires à l'aide du caractère **#**. Chaque enregistrement doit être séparé du précédent par une ligne blanche et il ne peut pas avoir deux lignes blanches consécutives.

Les attributs peuvent être sur plusieurs lignes. Dans ce cas les lignes supplémentaires commencent par un blanc. Par exemple :

```
dn: o=fenestros.loc
objectClass: dcObject
objectClass: organization
dc: fenestros
o: fenestros.loc
description: Exemple

dn: ou=utilisateurs,o=fenestros.loc
objectClass: organizationalUnit
objectClass: top
ou: utilisateurs

dn: cn=admin3,ou=utilisateurs,o=fenestros.loc
objectClass: person
objectClass: top
cn: admin3
sn: admin3
userPassword: fenestros

dn: ou=roles,o=fenestros.loc
objectClass: organizationalUnit
objectClass: top
```

```
ou: roles

dn: cn=manager-script,ou=roles,o=fenestros.loc
objectClass: groupOfUniqueNames
objectClass: top
cn: manager-script
uniqueMember: cn=admin3,ou=utilisateurs,o=fenestros.loc
```

La commande ldapadd

Afin de pouvoir utiliser notre fichier LDIF, il est nécessaire de faire appel au client **ldapadd**. Cet utilitaire prend un ou plusieurs options :

```
[root@centos7 bin]# ldapadd --help
ldapadd: invalid option -- '-'
ldapadd: unrecognized option --
Add or modify entries from an LDAP server

usage: ldapadd [options]
    The list of desired operations are read from stdin or from the file
    specified by "-f file".
Add or modify options:
  -a         add values (default)
  -c         continuous operation mode (do not stop on errors)
  -E [!]ext=extparam    modify extensions (! indicate s criticality)
  -f file    read operations from `file'
  -M         enable Manage DSA IT control (-MM to make critical)
  -P version protocol version (default: 3)
  -S file    write skipped modifications to `file'
Common options:
  -d level   set LDAP debugging level to `level'
  -D binddn  bind DN
  -e [!]<ext>[=<extparam>] general extensions (! indicates criticality)
             [!]assert=<filter>     (RFC 4528; a RFC 4515 Filter string)
```

```
            [!]authzid=<authzid>   (RFC 4370; "dn:<dn>" or "u:<user>")
            [!]chaining[=<resolveBehavior>[/<continuationBehavior>]]
                    one of "chainingPreferred", "chainingRequired",
                    "referralsPreferred", "referralsRequired"
            [!]manageDSAit         (RFC 3296)
            [!]noop
            ppolicy
            [!]postread[=<attrs>]  (RFC 4527; comma-separated attr list)
            [!]preread[=<attrs>]   (RFC 4527; comma-separated attr list)
            [!]relax
            [!]sessiontracking
            abandon, cancel, ignore (SIGINT sends abandon/cancel,
            or ignores response; if critical, doesn't wait for SIGINT.
            not really controls)
  -h host   LDAP server
  -H URI    LDAP Uniform Resource Identifier(s)
  -I        use SASL Interactive mode
  -n        show what would be done but don't actually do it
  -N        do not use reverse DNS to canonicalize SASL host name
  -O props  SASL security properties
  -o <opt>[=<optparam>] general options
            nettimeout=<timeout> (in seconds, or "none" or "max")
            ldif-wrap=<width> (in columns, or "no" for no wrapping)
  -p port   port on LDAP server
  -Q        use SASL Quiet mode
  -R realm  SASL realm
  -U authcid SASL authentication identity
  -v        run in verbose mode (diagnostics to standard output)
  -V        print version info (-VV only)
  -w passwd  bind password (for simple authentication)
  -W        prompt for bind password
  -x        Simple authentication
  -X authzid SASL authorization identity ("dn:<dn>" or "u:<user>")
  -y file   Read password from file
```

```
  -Y mech    SASL mechanism
  -Z         Start TLS request (-ZZ to require successful response)
```

Créez maintenant le fichier **/etc/openldap/slapd.conf** :

```
include     /etc/openldap/schema/corba.schema
include     /etc/openldap/schema/core.schema
include     /etc/openldap/schema/cosine.schema
include     /etc/openldap/schema/duaconf.schema
include     /etc/openldap/schema/dyngroup.schema
include     /etc/openldap/schema/inetorgperson.schema
include     /etc/openldap/schema/java.schema
include     /etc/openldap/schema/misc.schema
include     /etc/openldap/schema/nis.schema
include     /etc/openldap/schema/openldap.schema
include     /etc/openldap/schema/ppolicy.schema
include     /etc/openldap/schema/collective.schema
allow bind_v2
pidfile     /var/run/openldap/slapd.pid
argsfile    /var/run/openldap/slapd.args
TLSCACertificatePath /etc/openldap/certs
TLSCertificateFile "\"OpenLDAP Server\""
TLSCertificateKeyFile /etc/openldap/certs/password
database config
access to *
    by dn.exact="gidNumber=0+uidNumber=0,cn=peercred,cn=external,cn=auth" manage
    by * none
database monitor
access to *
    by dn.exact="gidNumber=0+uidNumber=0,cn=peercred,cn=external,cn=auth" read
        by dn.exact="cn=Manager,dc=my-domain,dc=com" read
        by * none
database    bdb
suffix          "o=fenestros.loc"
```

```
checkpoint  1024 15
rootdn          "cn=Manager,o=fenestros.loc"
rootpw      fenestros
directory   /var/lib/ldap
index objectClass                       eq,pres
index ou,cn,mail,surname,givenname      eq,pres,sub
index uidNumber,gidNumber,loginShell    eq,pres
index uid,memberUid                     eq,pres,sub
index nisMapName,nisMapEntry            eq,pres,sub
```

Nettoyez les anciens fichiers de configuration et fichiers de données :

```
[root@centos7 ~]# rm -Rf /etc/openldap/slapd.d/*
[root@centos7 ~]# rm -f /var/lib/ldap/alock
[root@centos7 ~]# rm -f /var/lib/ldap/__db.00?
```

Copiez le fichier /usr/share/openldap-servers/DB_CONFIG.example vers **/var/lib/ldap/DB_CONFIG** :

```
[root@centos7 ~]# cp /usr/share/openldap-servers/DB_CONFIG.example /var/lib/ldap/DB_CONFIG
```

Initialisez ensuite l'arborescence dans **/etc/openldap/slapd.d** :

```
[root@centos7 ~]# slaptest -f /etc/openldap/slapd.conf -F /etc/openldap/slapd.d
config file testing succeeded
```

Vérifiez que l'arborescence initiale soit créée :

```
[root@centos7 ~]# ls -l /etc/openldap/slapd.d
total 8
drwxr-x--- 3 root root 4096 Nov  5 13:20 cn=config
-rw------- 1 root root 1258 Nov  5 13:20 cn=config.ldif
```

Modifiez le propriétaire, le groupe ainsi que le droits du répertoire **/etc/openldap/slapd.d** :

```
[root@centos7 ~]# chown -R ldap:ldap /etc/openldap/slapd.d
[root@centos7 ~]# chmod -R u+rwX /etc/openldap/slapd.d
```

Modifiez le propriétaire et le groupe répertoire **/var/lib/ldap/** ainsi que le fichier **/etc/openldap/slapd.conf** :

```
[root@centos7 ~]# chown -R ldap:ldap /var/lib/ldap/ /etc/openldap/slapd.conf
```

Redémarrez le service slapd :

```
[root@centos7 bin]# systemctl restart slapd.service
```

Créez maintenant notre fichier LDIF dans le répertoire /tmp :

```
[root@centos7 bin]# cd /tmp
[root@centos7 tmp]# vi setup.ldif
[root@centos7 tmp]# cat setup.ldif
dn: o=fenestros.loc
objectClass: dcObject
objectClass: organization
dc: fenestros
o: fenestros.loc
description: Exemple

dn: ou=utilisateurs,o=fenestros.loc
objectClass: organizationalUnit
objectClass: top
ou: utilisateurs

dn: cn=admin3,ou=utilisateurs,o=fenestros.loc
objectClass: person
objectClass: top
cn: admin3
sn: admin3
```

```
userPassword: fenestros

dn: ou=roles,o=fenestros.loc
objectClass: organizationalUnit
objectClass: top
ou: roles

dn: cn=manager-script,ou=roles,o=fenestros.loc
objectClass: groupOfUniqueNames
objectClass: top
cn: manager-script
uniqueMember: cn=admin3,ou=utilisateurs,o=fenestros.loc
```

Il convient maintenant d'utiliser la commande ldapadd afin d'injecter le contenu du fichier setup.ldif dans notre base :

```
[root@centos7 tmp]# ldapadd -f setup.ldif -x -D "cn=Manager,o=fenestros.loc" -w fenestros
adding new entry "o=fenestros.loc"

adding new entry "ou=utilisateurs,o=fenestros.loc"

adding new entry "cn=admin3,ou=utilisateurs,o=fenestros.loc"

adding new entry "ou=roles,o=fenestros.loc"

adding new entry "cn=manager-script,ou=roles,o=fenestros.loc"
```

L'arborescence LDAP est la suivante :

```
localhost
        |
    o=fenestros.loc
        |
        ou=roles
        |    |
```

```
        |    cn=manager-script
        |
        ou=users
            |
            cn=admin3
```

Ajoutez ensuite la section suivante au fichier **$CATALINA_HOME/conf/server.xml** en mettant en commentaires le <Realm> précédent :

```
...
<!-- <Realm className="org.apache.catalina.realm.UserDatabaseRealm"
               resourceName="UserDatabase" digest="sha" /> -->

<!--  <Realm  className="org.apache.catalina.realm.JDBCRealm"
             driverName="com.mysql.jdbc.Driver"
          connectionURL="jdbc:mysql://localhost:3306/auth_tomcat"
         connectionName="tomcat" connectionPassword="secret"
              userTable="users" userNameCol="nom_user" userCredCol="mdp_user"
          userRoleTable="roles" roleNameCol="nom_role" />

      </Realm> -->

<!--   <Realm  className="org.apache.catalina.realm.DataSourceRealm"
         dataSourceName="jdbc/AuthTomcat"
              userTable="users" userNameCol="nom_user" userCredCol="mdp_user"
           userRoleTable="roles" roleNameCol="nom_role" />
    </Realm> -->

      <Realm  className="org.apache.catalina.realm.JNDIRealm"
          connectionURL="ldap://localhost:389"
         connectionName="cn=Manager,o=fenestros.loc"
     connectionPassword="fenestros"
               roleBase="ou=roles,o=fenestros.loc"
           roleName="cn"
         roleSearch="(uniqueMember={0})"
```

```
           userPassword="userPassword"
            userPattern="cn={0},ou=utilisateurs,o=fenestros.loc" />
    </Realm>

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

Redémarrez ensuite le service tomcat :

```
[root@centos7 tmp]# cd $CATALINA_HOME/bin
[root@centos7 bin]# ./shutdown.sh
Using CATALINA_BASE:   /usr/tomcat8
Using CATALINA_HOME:   /usr/tomcat8
Using CATALINA_TMPDIR: /usr/tomcat8/temp
Using JRE_HOME:        /usr/lib/jvm/jre-1.8.0-openjdk-1.8.0.232.b09-0.el7_7.x86_64
Using CLASSPATH:       /usr/tomcat8/bin/bootstrap.jar:/usr/tomcat8/bin/tomcat-juli.jar
[root@centos7 bin]# ./startup.sh
Using CATALINA_BASE:   /usr/tomcat8
Using CATALINA_HOME:   /usr/tomcat8
Using CATALINA_TMPDIR: /usr/tomcat8/temp
Using JRE_HOME:        /usr/lib/jvm/jre-1.8.0-openjdk-1.8.0.232.b09-0.el7_7.x86_64
Using CLASSPATH:       /usr/tomcat8/bin/bootstrap.jar:/usr/tomcat8/bin/tomcat-juli.jar
Tomcat started.
```

Vérifiez maintenant que vous pouvez vous connecter avec le compte **admin3** :

```
[root@centos7 bin]# lynx --dump -auth admin3:fenestros "http://www.i2tch.loc:8080/manager/text/serverinfo"
OK - Server info
Tomcat Version: Apache Tomcat/8.0.36
OS Name: Linux
OS Version: 3.10.0-1062.4.1.el7.x86_64
OS Architecture: amd64
JVM Version: 1.8.0_232-b09
```

```
JVM Vendor: Oracle Corporation
```

JAAS Realm

Le Realm JAAS est utilisé pour authentifier un utilisateur contre n'importe quel registre de stockage d'informations en développant un module d'authentification adéquat pour le registre concerné.

L'étude du développement d'un tel module dépasse le cadre de cette formation.

Combined Realm

Ce type de Realm utilise la classe Java **org.apache.catalina.realm.CombinedRealm**. Il permet de chaîner plusieurs Realms pour l'authentification afin de fournir une solution de disponibilité sur l'authentification.

Ce Realm ne contient qu'un seul attribut : **className**.

Voici un **exemple** de la section Realm du fichier $CATALINA_HOME/conf/server.xml :

```
    <Realm className="org.apache.catalina.realm.CombinedRealm">
            <Realm className="org.apache.catalina.realm.UserDatabaseRealm"
            resourceName="UserDatabase_1"/>
            <Realm className="org.apache.catalina.realm.UserDatabaseRealm"
            resourceName="UserDatabase_2"/>
    </Realm>
```

Important : L'authentification d'un seul des ces sous-Realms est suffisante pour autoriser l'utilisateur.

LockOut Realm

Ce type de Realm utilise la classe Java **org.apache.catalina.realm.LockOutRealm**. Il permet d'appliquer des des règles de blocage de comptes pour les sous-Realms qu'il englobe.

Ce Realm contient trois attributs :

- **className**,
- **failureCount**,
  - Spécifie le nombre de tentatives echouées de connexion avant un blocage. Par défaut la valeur est de **5**.
- **lockOutTime**,
  - Spécifie le nombre de secondes que le compte est bloqué. Par défaut la valeur est de **300**.

Voici un **exemple** de la section Realm du fichier $CATALINA_HOME/conf/server.xml :

```
    <Realm className="org.apache.catalina.realm.LockOutRealm">
            <Realm className="org.apache.catalina.realm.UserDatabaseRealm"
            resourceName="UserDatabase"/>
    </Realm>
```

Tomcat et le SSO

Le comportement par défaut de Tomcat est de demander une authentification par application.

Dans le cas où on souhaite mettre en place un **Single Sign-On**, il convient d'utiliser un élément <Valve>.

La configuration est la suivante :

```
<Host name ="localhost" ...>
  ...
  <Valve className="org.apache.catalina.authenticator.SingleSignOn" debug="0" />
  ...
```

Tomcat et le SSL

Présentation de SSL

SSL (*Secure Sockets Layers*) est utilisé pour sécuriser des transactions effectuées sur le Web et a été mis au point par :

- Netscape
- MasterCard
- Bank of America
- MCI
- Silicon Graphics

SSL est indépendant du protocole utilisé et agit en tant que couche supplémentaire entre la couche Application et la couche Transport. Il peut être utilisé avec :

- HTTP
- FTP
- POP
- IMAP

Fonctionnement de SSL

Le fonctionnement de SSL suit la procédure suivante :

- Le navigateur demande une page web sécurisée en https,
- Le serveur web émet sa clé publique et son certificat,
- Le navigateur vérifie que le certificat a été émis par une autorité fiable, qu'il est valide et qu'il fait référence au site consulté,
- Le navigateur utilise la clé publique du serveur pour chiffrer une clé symétrique aléatoire, une clé de session, et l'envoie au serveur avec l'URL demandé ainsi que des données HTTP chiffrées,
- Le serveur déchiffre la clé symétrique avec sa clé privée et l'utilise pour récupérer l'URL demandé et les données HTTP,
- Le serveur renvoie le document référencé par l'URL ainsi que les données HTTP chiffrées avec la clé symétrique,
- Le navigateur déchiffre le tout avec la clé symétrique et affiche les informations.

Quand on parle de **SSL**, on parle de **cryptologie**.

Configurer Tomcat

Pour utiliser SSL avec Tomcat, il est nécessaire d'avoir un répertoire pour stocker le fichier **.keystore**. Ce fichier doit contenir le certificat du serveur.

Commencez donc par créer ce répertoire :

```
[root@centos7 bin]# mkdir $CATALINA_HOME/security
```

Pour générer le certificat, il faut d'abord créer une clef privée. Cette clef est créée par la commande **keytool** :

```
[root@centos7 bin]# keytool -genkey -alias tomcat -keyalg RSA -keystore
$CATALINA_HOME/security/www_i2tch_loc.keystore
Enter keystore password: fenestros
Re-enter new password: fenestros
What is your first and last name?
  [Unknown]:  HUGH NORRIS
What is the name of your organizational unit?
  [Unknown]:  TRAINING
What is the name of your organization?
  [Unknown]:  I2TCH LIMITED
What is the name of your City or Locality?
  [Unknown]:  ADDLESTON
What is the name of your State or Province?
  [Unknown]:  SURREY
What is the two-letter country code for this unit?
  [Unknown]:  GB
Is CN=HUGH NORRIS, OU=TRAINING, O=I2TCH LIMITED, L=ADDLESTON, ST=SURREY, C=GB correct?
  [no]:  YES

Enter key password for <tomcat>
    (RETURN if same as keystore password): [Return]
```

```
Warning:
The JKS keystore uses a proprietary format. It is recommended to migrate to PKCS12 which is an industry standard
format using "keytool -importkeystore -srckeystore /usr/tomcat8/security/www_i2tch_loc.keystore -destkeystore
/usr/tomcat8/security/www_i2tch_loc.keystore -deststoretype pkcs12".
```

Après la création de la clef, il est nécessaire de créer un **CSR** (*Certificate Signing Request*). Pour créer le CSR, il convient d'utiliser de nouveau la commande **keytool** :

```
[root@centos7 bin]# keytool -certreq -keyalg RSA -alias tomcat -file www_i2tch_loc.csr -keystore
$CATALINA_HOME/security/www_i2tch_loc.keystore
Enter keystore password:  fenestros

Warning:
The JKS keystore uses a proprietary format. It is recommended to migrate to PKCS12 which is an industry standard
format using "keytool -importkeystore -srckeystore /usr/tomcat8/security/www_i2tch_loc.keystore -destkeystore
/usr/tomcat8/security/www_i2tch_loc.keystore -deststoretype pkcs12".
```

A ce stade, vous enverriez votre CSR à un organisme PKI tel **VeriSign** qui, après vérification des informations contenues dans votre CRT, signerait votre demande avec leur clef produisant ainsi un certificat qu'il vous retourne accompagné de son **Certificat Racine**.

Après réception de ce fichier, vous devez importer le **Certificat Racine** de votre PKI dans votre keystore, **par exemple** :

```
# keytool -import -alias root -keystore $CATALINA_HOME/security/www_i2tch_loc.keystore -file
<nom_du_certificat_racine>
```

Ensuite, vous devez importer votre propre certificat, **par exemple** :

```
# keytool -import -alias tomcat -keystore $CATALINA_HOME/security/www_i2tch_loc.keystore -trustcacerts -file
<nom_de_votre_certificat>
```

Pour plus d'informations concernant la création d'un keystore au format **.jks**, consultez cette page.

Dans le cas de ce LAB, nous n'allons pas faire appelle à un PKI. Par conséquent, il convient de signé notre propre CRT avec notre clef privée. Cette action génère un certificat SSL directement dans le keystore :

```
[root@centos7 bin]# keytool -selfcert -alias tomcat -keypass fenestros -keystore
$CATALINA_HOME/security/www_i2tch_loc.keystore -dname "CN=HUGH NORRIS, OU=TRAINING, O=I2TCH LIMITED, L=ADDLESTON,
ST=SURREY, C=GB"
Enter keystore password:  fenestros

Warning:
The JKS keystore uses a proprietary format. It is recommended to migrate to PKCS12 which is an industry standard
format using "keytool -importkeystore -srckeystore /usr/tomcat8/security/www_i2tch_loc.keystore -destkeystore
/usr/tomcat8/security/www_i2tch_loc.keystore -deststoretype pkcs12".
```

Dernièrement, ajoutez le connector suivant au fichier **$CATALINA_HOME/conf/server.xml** :

```
...

    <Connector executor="tomcatThreadPool"
               port="8080" protocol="HTTP/1.1"
               connectionTimeout="20000"
               redirectPort="8443" />

    <!-- Define a SSL/TLS HTTP/1.1 Connector on port 8443
         This connector uses the NIO implementation that requires the JSSE
         style configuration. When using the APR/native implementation, the
         OpenSSL style configuration is required as described in the APR/native
         documentation -->
    <!--
    <Connector port="8443" protocol="org.apache.coyote.http11.Http11NioProtocol"
               maxThreads="150" SSLEnabled="true" scheme="https" secure="true"
               clientAuth="false" sslProtocol="TLS" />
    -->

    <Connector port="8443" maxThreads="150" minSpareThreads="25" maxSpareThreads="75"
               enableLookups="false" disableUploadTimeout="true"
               acceptCount="100" debug="0" SSLEnabled="true" scheme="https" secure="true"
               clientAuth="false" sslProtocol="TLS"
```

```
               keystoreFile="/usr/tomcat8/security/www_i2tch_loc.keystore"
               keystorePass="fenestros" />
...
```

Redémarrez le serveur Tomcat :

```
[root@centos7 bin]# ./shutdown.sh
Using CATALINA_BASE:   /usr/tomcat8
Using CATALINA_HOME:   /usr/tomcat8
Using CATALINA_TMPDIR: /usr/tomcat8/temp
Using JRE_HOME:        /usr/lib/jvm/jre-1.8.0-openjdk-1.8.0.232.b09-0.el7_7.x86_64
Using CLASSPATH:       /usr/tomcat8/bin/bootstrap.jar:/usr/tomcat8/bin/tomcat-juli.jar
[root@centos7 bin]# ./startup.sh
Using CATALINA_BASE:   /usr/tomcat8
Using CATALINA_HOME:   /usr/tomcat8
Using CATALINA_TMPDIR: /usr/tomcat8/temp
Using JRE_HOME:        /usr/lib/jvm/jre-1.8.0-openjdk-1.8.0.232.b09-0.el7_7.x86_64
Using CLASSPATH:       /usr/tomcat8/bin/bootstrap.jar:/usr/tomcat8/bin/tomcat-juli.jar
Tomcat started.
```

Vérifiez maintenant que vous pouvez vous connecter avec le compte **admin3** sur le port **8443** en utilisant le navigateur Firefox et le lien **https://www.i2tch.loc:8443/manager/text/serverinfo**.

Configurer Apache

Installation de SSL

Installez le module **mod_ssl** pour Apache :

```
[root@centos7 bin]# rpm -qa | grep ssl
python-backports-ssl_match_hostname-3.5.0.1-1.el7.noarch
openssl-1.0.2k-19.el7.x86_64
```

```
openssl-libs-1.0.2k-19.el7.x86_64
xmlsec1-openssl-1.2.20-7.el7_4.x86_64
[root@centos7 bin]#
[root@centos7 bin]# yum install mod_ssl
...
[root@centos7 bin]# rpm -qa | grep ssl
python-backports-ssl_match_hostname-3.5.0.1-1.el7.noarch
openssl-1.0.2k-19.el7.x86_64
openssl-libs-1.0.2k-19.el7.x86_64
xmlsec1-openssl-1.2.20-7.el7_4.x86_64
mod_ssl-2.4.6-90.el7.centos.x86_64
```

Configuration de SSL

Dans le cas où vous souhaitez générer vos propres clés, vous devez d'abord générer une clé privée, nécessaire pour la création d'un **Certificate Signing Request**. Le CSR doit alors être envoyé à une des sociétés faisant autorité en la matière afin que celle-ci puisse vous retourner votre certificat définitif. Ce service est payant. C'est ce certificat définitif qui est utilisé pour des connexions sécurisées.

Saisissez donc la commande suivante pour générer votre clé privée :

```
[root@centos7 bin]# cd /root ; openssl genrsa -out www.i2tch.loc.key 1024
Generating RSA private key, 1024 bit long modulus
..............++++++
.........................................++++++
e is 65537 (0x10001)
```

Générer maintenant votre CSR :

```
[root@centos7 ~]# openssl req -new -key www.i2tch.loc.key -out www.i2tch.loc.csr
You are about to be asked to enter information that will be incorporated
into your certificate request.
What you are about to enter is what is called a Distinguished Name or a DN.
There are quite a few fields but you can leave some blank
```

```
For some fields there will be a default value,
If you enter '.', the field will be left blank.
-----
Country Name (2 letter code) [XX]:GB
State or Province Name (full name) []:SURREY
Locality Name (eg, city) [Default City]:ADDLESTON
Organization Name (eg, company) [Default Company Ltd]:I2TCH LIMITED
Organizational Unit Name (eg, section) []:TRAINING
Common Name (eg, your name or your server's hostname) []:www.i2tch.loc
Email Address []:infos@i2tch.co.uk

Please enter the following 'extra' attributes
to be sent with your certificate request
A challenge password []: [RETURN]
An optional company name []: [RETURN]
```

et répondez aux questions qui vous sont posées. Notez bien la réponse à la question **Common Name**. Si vous ne donnez pas le nom de votre site, certains navigateurs ne géreront pas votre certificat correctement. Vous pouvez maintenant envoyé votre CSR à la société que vous avez choisie. Quand votre clé **.crt** vous est retournée, copiez-la, ainsi que votre clé privée dans le répertoire **/etc/pki/tls/certs/**.

Sans passer par un prestataire externe, vous pouvez signer votre CSR avec votre propre clé afin de générer votre certificat :

```
[root@centos7 ~]# openssl x509 -req -days 365 -in www.i2tch.loc.csr -signkey www.i2tch.loc.key -out
www.i2tch.loc.crt
Signature ok
subject=/C=GB/ST=SURREY/L=ADDLESTON/O=I2TCH LIMITED/OU=TRAINING/CN=www.i2tch.loc/emailAddress=infos@i2tch.eu
Getting Private key
```

Il convient ensuite de copier le certificat dans le répertoire **/etc/pki/tls/certs/** et la clef privée dans le répertoire **/etc/pki/tls/private/** :

```
[root@centos7 ~]# cp /root/www.i2tch.loc.key /etc/pki/tls/private/
[root@centos7 ~]# cp /root/www.i2tch.loc.crt /etc/pki/tls/certs/
```

Mise en place des paramètres de sécurité SSL

Consultez le contenu du répertoire **/etc/httpd/conf.d/** :

```
[root@centos7 ~]# ls /etc/httpd/conf.d
autoindex.conf  README  ssl.conf  userdir.conf  welcome.conf
```

Ce répertoire contient des fichiers dont le contenu est inclut dans le corps du fichier httpd.conf.

Sauvegardez le fichier **ssl.conf**.

```
[root@centos7 ~]# cp /etc/httpd/conf.d/ssl.conf /root/ssl.conf.backup
```

Ouvrez le fichier /etc/httpd/conf.d/ssl.conf et modifiez la ligne suivante :

```
#DocumentRoot "/var/www/html"
```

en :

```
DocumentRoot "/var/www/html"
```

Cette directive indique que la racine du site sécurisé sera **/var/www/html**.

Deuxièmement, ajoutez la ligne suivante en dessous de la directive **#ServerName …** existante :

```
ServerName www.i2tch.loc:443
```

Dernièrement modifiez les deux lignes suivantes :

```
SSLCertificateFile /etc/pki/tls/certs/localhost.crt
```

```
SSLCertificateKeyFile /etc/pki/tls/private/localhost.key
```

en :

```
SSLCertificateFile /etc/pki/tls/certs/www.i2tch.loc.crt
```

```
SSLCertificateKeyFile /etc/pki/tls/private/www.i2tch.loc.key
```

Vous obtiendrez :

```
...
#   Server Certificate:
# Point SSLCertificateFile at a PEM encoded certificate.  If
# the certificate is encrypted, then you will be prompted for a
# pass phrase.  Note that a kill -HUP will prompt again.  A new
# certificate can be generated using the genkey(1) command.
#SSLCertificateFile /etc/pki/tls/certs/localhost.crt
SSLCertificateFile /etc/pki/tls/certs/www.i2tch.loc.crt

#   Server Private Key:
#   If the key is not combined with the certificate, use this
#   directive to point at the key file.  Keep in mind that if
#   you've both a RSA and a DSA private key you can configure
#   both in parallel (to also allow the use of DSA ciphers, etc.)
# SSLCertificateKeyFile /etc/pki/tls/private/localhost.key
SSLCertificateKeyFile /etc/pki/tls/private/www.i2tch.loc.key
...
```

Sauvegardez le fichier et redémarrez le serveur apache :

```
[root@centos7 ~]# systemctl restart httpd.service
```

Tester Votre Configuration

Pour tester votre serveur apache en mode SSL, vous allez procéder à deux tests distincts.

Dans le premier, saisissez la commande suivante en ligne de commande et en tant que root :

```
[root@centos7 ~]# openssl s_client -connect www.i2tch.loc:443
CONNECTED(00000003)
depth=0 C = GB, ST = SURREY, L = ADDLESTON, O = I2TCH LIMITED, OU = TRAINING, CN = www.i2tch.loc, emailAddress =
infos@i2tch.eu
verify error:num=18:self signed certificate
verify return:1
depth=0 C = GB, ST = SURREY, L = ADDLESTON, O = I2TCH LIMITED, OU = TRAINING, CN = www.i2tch.loc, emailAddress =
infos@i2tch.eu
verify return:1
---
Certificate chain
 0 s:/C=GB/ST=SURREY/L=ADDLESTON/O=I2TCH LIMITED/OU=TRAINING/CN=www.i2tch.loc/emailAddress=infos@i2tch.eu
   i:/C=GB/ST=SURREY/L=ADDLESTON/O=I2TCH LIMITED/OU=TRAINING/CN=www.i2tch.loc/emailAddress=infos@i2tch.eu
---
Server certificate
-----BEGIN CERTIFICATE-----
MIICoTCCAgoCCQD2lg50Gvmh/jANBgkqhkiG9w0BAQUFADCBlDELMAkGA1UEBhMC
R0IxDzANBgNVBAgMBlNVUlJFWTESMBAGA1UEBwwJQURETEVTVE9OMRYwFAYDVQQK
DA1JMlRDSCBMSU1JVEVEMREwDwYDVQQLDAhUUkFJTklORzEWMBQGA1UEAwwNd3d3
LmkydGNoLmxvYzEdMBsGCSqGSIb3DQEJARYOaW5mb3NAaTJ0Y2guZXUwHhcNMTYw
NjI4MTYyNDMxWhcNMTcwNjI4MTYyNDMxWjCBlDELMAkGA1UEBhMCR0IxDzANBgNV
BAgMBlNVUlJFWTESMBAGA1UEBwwJQURETEVTVE9OMRYwFAYDVQQKDA1JMlRDSCBM
SU1JVEVEMREwDwYDVQQLDAhUUkFJTklORzEWMBQGA1UEAwwNd3d3LmkydGNoLmxv
YzEdMBsGCSqGSIb3DQEJARYOaW5mb3NAaTJ0Y2guZXUwgZ8wDQYJKoZIhvcNAQEB
BQADgY0AMIGJAoGBAKRfbHwr4FNll+cwYN7gkIIp30EhwOgZU6ql/4K49zw54Sk9
vZ2uhECcrtHcdVwGJtieLjDVWoCI4RIflv6XXvHdUbScpRKHadrQDtvySHAzLw9L
SDbOGj26zK8dHFUCE2lDVyVGVWHgvgdCtrYJxDjwqZlmCOqA9Ii5587h8McpAgMB
AAEwDQYJKoZIhvcNAQEFBQADgYEAJiGMZlGh8a+DfzXGI9fwoX+kS/nSNotS5D0j
NaVM97NCwAtH/DX4qUYUp1UDmIpCEaZcO8e9Xt/GDjkJRSb6SqufFREbs1kRfKzW
3ar2Xnnt0Qr73YXr03uMP+/WdmIDF9NawLZ7C37m+KgnUJQlLuE+nUMjeQI1ogYM
```

```
KdTjblI=
-----END CERTIFICATE-----
subject=/C=GB/ST=SURREY/L=ADDLESTON/O=I2TCH LIMITED/OU=TRAINING/CN=www.i2tch.loc/emailAddress=infos@i2tch.eu
issuer=/C=GB/ST=SURREY/L=ADDLESTON/O=I2TCH LIMITED/OU=TRAINING/CN=www.i2tch.loc/emailAddress=infos@i2tch.eu
---
No client certificate CA names sent
Server Temp Key: ECDH, secp521r1, 521 bits
---
SSL handshake has read 1308 bytes and written 441 bytes
---
New, TLSv1/SSLv3, Cipher is ECDHE-RSA-AES256-GCM-SHA384
Server public key is 1024 bit
Secure Renegotiation IS supported
Compression: NONE
Expansion: NONE
SSL-Session:
    Protocol  : TLSv1.2
    Cipher    : ECDHE-RSA-AES256-GCM-SHA384
    Session-ID: C8B3E6709FCB21D5014F0938AA73F32E1D3F73A6B38FF3A238032CD268808916
    Session-ID-ctx:
    Master-Key: 798987A0DADE2C89E7915B789A14979A3FDC4D9371A37A916273CE98890DB9978B401072867053FC9178B3453F219360
    Key-Arg   : None
    Krb5 Principal: None
    PSK identity: None
    PSK identity hint: None
    TLS session ticket lifetime hint: 300 (seconds)
    TLS session ticket:
    0000 - 40 e7 3a 85 fe 5f e5 a2-60 cd 51 4e 1b 49 f9 d5   @.:.._..`.QN.I..
    0010 - b3 65 8c 63 72 b3 60 1c-47 ed 83 5e ae 42 55 da   .e.cr.`.G..^.BU.
    0020 - f6 7f 30 1d 3b 97 a8 09-31 25 35 58 24 f4 ee a3   ..0.;...1%5X$...
    0030 - d7 a1 19 30 be e5 6e 93-4a ec 19 ae c0 49 85 5b   ...0..n.J....I.[
    0040 - 5c 0f bf 04 f8 31 b1 92-99 0c f3 cf fd 85 42 83   \....1........B.
    0050 - 07 2f 31 21 c9 f1 44 19-57 1a 6c c4 20 dc 5f 22   ./1!..D.W.l. ._"
    0060 - 11 6d ac cf 60 fc 68 20-4e ac 03 f7 55 f6 31 25   .m..`.h N...U.1%
```

```
    0070 - 64 91 dc fc be 02 47 0d-46 a5 ad cd 7b a4 69 41   d.....G.F...{.iA
    0080 - 5c 32 25 cf fe 1c af cb-fc 7a 0d 33 47 e0 43 93   \2%......z.3G.C.
    0090 - b3 57 73 e1 69 30 cd 25-1e 0d b8 74 13 66 93 01   .Ws.i0.%...t.f..
    00a0 - aa 29 3d 25 32 59 4e ad-3d 00 ca 26 48 7b f0 bc   .)=%2YN.=..&H{..
    00b0 - 30 8c ed d9 e9 ae 23 26-f1 a4 ee 35 91 16 82 ff   0.....#&...5....

    Start Time: 1467131982
    Timeout   : 300 (sec)
    Verify return code: 18 (self signed certificate)
---
^C
```

Notez qu'il y a génération d'erreurs. Ceci est normal. Ce test démontre que votre site sécurisé fonctionne. Votre serveur apache a été configuré avec succès.

Apache en Frontal HTTPS

Pour utiliser le serveur web Apache en tant que serveur frontal HTTPS pour Tomcat, il convient d'utiliser le proxy d'apache. Vérifiez donc que les deux lignes **LoadModule proxy_module modules/mod_proxy.so** et **LoadModule proxy_http_module modules/mod_proxy_http.so** soient bien présentes dans le fichier **/etc/httpd/conf.modules.d/00-proxy.conf** :

```
[root@centos7 ~]# cat /etc/httpd/conf.modules.d/00-proxy.conf
# This file configures all the proxy modules:
LoadModule proxy_module modules/mod_proxy.so
LoadModule lbmethod_bybusyness_module modules/mod_lbmethod_bybusyness.so
LoadModule lbmethod_byrequests_module modules/mod_lbmethod_byrequests.so
LoadModule lbmethod_bytraffic_module modules/mod_lbmethod_bytraffic.so
LoadModule lbmethod_heartbeat_module modules/mod_lbmethod_heartbeat.so
LoadModule proxy_ajp_module modules/mod_proxy_ajp.so
LoadModule proxy_balancer_module modules/mod_proxy_balancer.so
LoadModule proxy_connect_module modules/mod_proxy_connect.so
LoadModule proxy_express_module modules/mod_proxy_express.so
LoadModule proxy_fcgi_module modules/mod_proxy_fcgi.so
```

```
LoadModule proxy_fdpass_module modules/mod_proxy_fdpass.so
LoadModule proxy_ftp_module modules/mod_proxy_ftp.so
LoadModule proxy_http_module modules/mod_proxy_http.so
LoadModule proxy_scgi_module modules/mod_proxy_scgi.so
LoadModule proxy_wstunnel_module modules/mod_proxy_wstunnel.so
```

Ajoutez maintenant les directives suivantes à la fin du fichier **/etc/httpd/conf.d/ssl.conf** juste avant la balise </VirtualHost> :

```
[root@centos7 ~]# cat /etc/httpd/conf.d/ssl.conf
...
#   Per-Server Logging:
#   The home of a custom SSL log file. Use this when you want a
#   compact non-error SSL logfile on a virtual host basis.
CustomLog logs/ssl_request_log \
          "%t %h %{SSL_PROTOCOL}x %{SSL_CIPHER}x \"%r\" %b"
<IfModule mod_proxy.c>
    ProxyRequests                   Off
    ProxyPreserveHost               On
    ProxyPass           /docs       http://www.i2tch.loc:8443/docs
    ProxyPassReverse    /docs       http://www.i2tch.loc:8443/docs
</IfModule>

</VirtualHost>
```

Sauvegardez et relancez le service httpd :

```
[root@centos7 ~]# systemctl restart httpd.service
```

Éditez maintenant le Connector HTTPS du fichier **$CATALINA_HOME/conf/server.xml** :

```
...
    <!--
    <Connector port="8443" maxThreads="150" minSpareThreads="25" maxSpareThreads="75"
               enableLookups="false" disableUploadTimeout="true"
```

```
               acceptCount="100" debug="0" scheme="https" secure="true"
               clientAuth="false" sslProtocol="TLS"
               keystoreFile="/usr/tomcat8/security/www_i2tch_loc.keystore"
               keystorePass="fenestros" />
    -->

    <Connector port="8443" proxyPort="443" proxyName="10.0.2.51" />
...
```

Sauvegardez et relancez le service tomcat :

```
[root@centos7 ~]# cd $CATALINA_HOME/bin
[root@centos7 bin]# ./shutdown.sh
Using CATALINA_BASE:   /usr/tomcat8
Using CATALINA_HOME:   /usr/tomcat8
Using CATALINA_TMPDIR: /usr/tomcat8/temp
Using JRE_HOME:        /usr/lib/jvm/jre-1.8.0-openjdk-1.8.0.232.b09-0.el7_7.x86_64
Using CLASSPATH:       /usr/tomcat8/bin/bootstrap.jar:/usr/tomcat8/bin/tomcat-juli.jar
[root@centos7 bin]# ./startup.sh
Using CATALINA_BASE:   /usr/tomcat8
Using CATALINA_HOME:   /usr/tomcat8
Using CATALINA_TMPDIR: /usr/tomcat8/temp
Using JRE_HOME:        /usr/lib/jvm/jre-1.8.0-openjdk-1.8.0.232.b09-0.el7_7.x86_64
Using CLASSPATH:       /usr/tomcat8/bin/bootstrap.jar:/usr/tomcat8/bin/tomcat-juli.jar
Tomcat started.
```

A ce stade, le serveur Tomcat ne propose plus de <Connector> direct en https. Par contre, grâce à la configuration du proxy apache, les connexions à l'application **docs** seront cryptées par le SSL d'apache.

Afin de vous assurer que la configuration est bien faite, saisissez l'URL suivant dans le navigateur de votre VM : **https://www.i2tch.loc/docs**.

Restrictions d'Accès

Les restrictions d'accès de Tomcat permet la mise en place de restrictions pour :

- le Serveur,
- un hôte particulier de ce serveur,
- une application.

Les Valves utilisent les classes **org.apache.catalina.valves.RemoteHostValve** et **org.apache.catalina.valves.RemoteAddrValve**.

Les deux filtres ci-dessus utilisent les même attributs :

- **allow**,
- **deny**.

Les deux attributs prennent en tant que valeur une expression régulière au format **java.util.regex** identifiant des adresses IP ou des noms d'hôtes.

Le dernier attribut est **denyStatus** qui permet de spécifier quel code d'erreur HTML sera envoyé vers le navigateur du client. Par défaut cette valeur est **403** *Forbidden*.

Voici deux exemples de restrictions :

```
    <Valve className='org.apache.catelina.valves.RemoteAddrValve"
        allow="127\.0\.0\.1|10\.\d+\.\d+" />
    </Valve>
```

```
    <Valve className='org.apache.catelina.valves.RemoteHostValve"
        allow="\w+\.i2tch\.loc" />
    </Valve>
```

Le Gestionnaire de Sécurité

La machine virtuelle Java dispose d'une classe Java spéciale appelée **SecurityManager** (*Gestionnaire de Sécurité*). Cette classe permet de bloquer

l'exécution de certaines classes ainsi que de bloquer l'accès à des ressources système aux classes qui s'exécutent. Une fois activé le Gestionnaire de Sécurité interdit tout en utilisant des fichiers de configuration qui se terminent en général par l'extension **.policy**.

Une directive d'un fichier .policy prend une syntaxe particulière :

```
grant [codeBase <code>] {
    permission <classe> [<nom>, <liste permissions>];
};
```

Par exemple pour autoriser **uniquement** la lecture du fichier **/tmp/fichier** par tout le code du répertoire **$JAVA_HOME/lib/ext**, la directive devient :

```
grant codeBase "file:${java.home}/lib/ext/*" {
    permission java.io.FilePermission "/tmp/fichier", "read";
};
```

Dans cette directive, la portée du codeBase diffère selon l'écriture de la clause **file:** :

- **file:${java.home}/lib/ext/**,
  - concerne uniquement les classes dans ${java.home}/lib/ext/ mais pas les classes et fichiers JAR des sous-répertoires,
- **file:${java.home}/lib/ext/*** ,
  - concerne les classes et le fichiers JAR dans ${java.home}/lib/ext/ mais pas les classes et fichiers JAR des sous-répertoires,
- **file:${java.home}/lib/ext/-**,
  - * concerne les classes et le fichiers JAR dans ${java.home}/lib/ext/ et celles et ceux dans des sous-répertoires.

> **Important** : Il est aussi possible d'utiliser **http:** à la place de **file:**.

La liste des permissions définies en standard est :

- **java.security.AllPermission**,
- **java.security.SecurityPermission**,
- **java.io.SerializablePermission**,
- **java.lang.reflect.ReflectPermission**,

- **java.lang.RuntimePermission**,
- **java.net.NetPermission**,
- **java.net.SocketPermission**,
- **java.util.PropertyPermission**.

Le fichier .policy chargé par défaut lors de l'activation du Gestionnaire de Sécurité est **$JAVA_HOME/lib/security/java.policy** :

```
[root@centos7 bin]# cat $JAVA_HOME/lib/security/java.policy

// Standard extensions get all permissions by default

grant codeBase "file:${{java.ext.dirs}}/*" {
        permission java.security.AllPermission;
};

// default permissions granted to all domains

grant {
        // Allows any thread to stop itself using the java.lang.Thread.stop()
        // method that takes no argument.
        // Note that this permission is granted by default only to remain
        // backwards compatible.
        // It is strongly recommended that you either remove this permission
        // from this policy file or further restrict it to code sources
        // that you specify, because Thread.stop() is potentially unsafe.
        // See the API specification of java.lang.Thread.stop() for more
        // information.
        permission java.lang.RuntimePermission "stopThread";

        // allows anyone to listen on dynamic ports
        permission java.net.SocketPermission "localhost:0", "listen";

        // "standard" properies that can be read by anyone
```

```
        permission java.util.PropertyPermission "java.version", "read";
        permission java.util.PropertyPermission "java.vendor", "read";
        permission java.util.PropertyPermission "java.vendor.url", "read";
        permission java.util.PropertyPermission "java.class.version", "read";
        permission java.util.PropertyPermission "os.name", "read";
        permission java.util.PropertyPermission "os.version", "read";
        permission java.util.PropertyPermission "os.arch", "read";
        permission java.util.PropertyPermission "file.separator", "read";
        permission java.util.PropertyPermission "path.separator", "read";
        permission java.util.PropertyPermission "line.separator", "read";

        permission java.util.PropertyPermission "java.specification.version", "read";
        permission java.util.PropertyPermission "java.specification.vendor", "read";
        permission java.util.PropertyPermission "java.specification.name", "read";

        permission java.util.PropertyPermission "java.vm.specification.version", "read";
        permission java.util.PropertyPermission "java.vm.specification.vendor", "read";
        permission java.util.PropertyPermission "java.vm.specification.name", "read";
        permission java.util.PropertyPermission "java.vm.version", "read";
        permission java.util.PropertyPermission "java.vm.vendor", "read";
        permission java.util.PropertyPermission "java.vm.name", "read";
};
```

Pour activer le Gestionnaire de Sécurité, il faut démarrer la machine virtuelle Java avec l'option **-Djava.security.manager**. Ceci est déjà prévu sous Tomcat. En effet il suffit de passer l'option **-security** au script **$CATALINA_HOME/bin/startup.sh**. Dans ce cas c'est le contenu du fichier **$CATALINA_HOME/conf/catalina.policy** qui est appliqué :

```
[root@centos7 bin]# cat $CATALINA_HOME/conf/catalina.policy
// Licensed to the Apache Software Foundation (ASF) under one or more
// contributor license agreements.  See the NOTICE file distributed with
// this work for additional information regarding copyright ownership.
// The ASF licenses this file to You under the Apache License, Version 2.0
// (the "License"); you may not use this file except in compliance with
// the License.  You may obtain a copy of the License at
```

```
//
//     http://www.apache.org/licenses/LICENSE-2.0
//
// Unless required by applicable law or agreed to in writing, software
// distributed under the License is distributed on an "AS IS" BASIS,
// WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
// See the License for the specific language governing permissions and
// limitations under the License.

// ============================================================================
// catalina.policy - Security Policy Permissions for Tomcat
//
// This file contains a default set of security policies to be enforced (by the
// JVM) when Catalina is executed with the "-security" option.  In addition
// to the permissions granted here, the following additional permissions are
// granted to each web application:
//
// * Read access to the web application's document root directory
// * Read, write and delete access to the web application's working directory
// ============================================================================


// ========== SYSTEM CODE PERMISSIONS =========================================


// These permissions apply to javac
grant codeBase "file:${java.home}/lib/-" {
        permission java.security.AllPermission;
};

// These permissions apply to all shared system extensions
grant codeBase "file:${java.home}/jre/lib/ext/-" {
        permission java.security.AllPermission;
};
```

```
// These permissions apply to javac when ${java.home] points at $JAVA_HOME/jre
grant codeBase "file:${java.home}/../lib/-" {
        permission java.security.AllPermission;
};

// These permissions apply to all shared system extensions when
// ${java.home} points at $JAVA_HOME/jre
grant codeBase "file:${java.home}/lib/ext/-" {
        permission java.security.AllPermission;
};


// ========== CATALINA CODE PERMISSIONS =======================================


// These permissions apply to the daemon code
grant codeBase "file:${catalina.home}/bin/commons-daemon.jar" {
        permission java.security.AllPermission;
};

// These permissions apply to the logging API
// Note: If tomcat-juli.jar is in ${catalina.base} and not in ${catalina.home},
// update this section accordingly.
//  grant codeBase "file:${catalina.base}/bin/tomcat-juli.jar" {..}
grant codeBase "file:${catalina.home}/bin/tomcat-juli.jar" {
        permission java.io.FilePermission
         "${java.home}${file.separator}lib${file.separator}logging.properties", "read";

        permission java.io.FilePermission
         "${catalina.base}${file.separator}conf${file.separator}logging.properties", "read";
        permission java.io.FilePermission
         "${catalina.base}${file.separator}logs", "read, write";
        permission java.io.FilePermission
         "${catalina.base}${file.separator}logs${file.separator}*", "read, write";
```

```
        permission java.lang.RuntimePermission "shutdownHooks";
        permission java.lang.RuntimePermission "getClassLoader";
        permission java.lang.RuntimePermission "setContextClassLoader";

        permission java.lang.management.ManagementPermission "monitor";

        permission java.util.logging.LoggingPermission "control";

        permission java.util.PropertyPermission "java.util.logging.config.class", "read";
        permission java.util.PropertyPermission "java.util.logging.config.file", "read";
        permission java.util.PropertyPermission "org.apache.juli.AsyncLoggerPollInterval", "read";
        permission java.util.PropertyPermission "org.apache.juli.AsyncMaxRecordCount", "read";
        permission java.util.PropertyPermission "org.apache.juli.AsyncOverflowDropType", "read";
        permission java.util.PropertyPermission "org.apache.juli.ClassLoaderLogManager.debug", "read";
        permission java.util.PropertyPermission "catalina.base", "read";

        // Note: To enable per context logging configuration, permit read access to
        // the appropriate file. Be sure that the logging configuration is
        // secure before enabling such access.
        // E.g. for the examples web application (uncomment and unwrap
        // the following to be on a single line):
        // permission java.io.FilePermission "${catalina.base}${file.separator}
        //  webapps${file.separator}examples${file.separator}WEB-INF
        //  ${file.separator}classes${file.separator}logging.properties", "read";
};

// These permissions apply to the server startup code
grant codeBase "file:${catalina.home}/bin/bootstrap.jar" {
        permission java.security.AllPermission;
};

// These permissions apply to the servlet API classes
// and those that are shared across all class loaders
// located in the "lib" directory
```

```
grant codeBase "file:${catalina.home}/lib/-" {
        permission java.security.AllPermission;
};


// If using a per instance lib directory, i.e. ${catalina.base}/lib,
// then the following permission will need to be uncommented
// grant codeBase "file:${catalina.base}/lib/-" {
//         permission java.security.AllPermission;
// };


// ========== WEB APPLICATION PERMISSIONS =====================================


// These permissions are granted by default to all web applications
// In addition, a web application will be given a read FilePermission
// for all files and directories in its document root.
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";

    // OS Specific properties to allow read access
    permission java.util.PropertyPermission "os.name", "read";
    permission java.util.PropertyPermission "os.version", "read";
    permission java.util.PropertyPermission "os.arch", "read";
    permission java.util.PropertyPermission "file.separator", "read";
    permission java.util.PropertyPermission "path.separator", "read";
    permission java.util.PropertyPermission "line.separator", "read";

    // JVM properties to allow read access
```

```
    permission java.util.PropertyPermission "java.version", "read";
    permission java.util.PropertyPermission "java.vendor", "read";
    permission java.util.PropertyPermission "java.vendor.url", "read";
    permission java.util.PropertyPermission "java.class.version", "read";
    permission java.util.PropertyPermission "java.specification.version", "read";
    permission java.util.PropertyPermission "java.specification.vendor", "read";
    permission java.util.PropertyPermission "java.specification.name", "read";

    permission java.util.PropertyPermission "java.vm.specification.version", "read";
    permission java.util.PropertyPermission "java.vm.specification.vendor", "read";
    permission java.util.PropertyPermission "java.vm.specification.name", "read";
    permission java.util.PropertyPermission "java.vm.version", "read";
    permission java.util.PropertyPermission "java.vm.vendor", "read";
    permission java.util.PropertyPermission "java.vm.name", "read";

    // Required for OpenJMX
    permission java.lang.RuntimePermission "getAttribute";

    // Allow read of JAXP compliant XML parser debug
    permission java.util.PropertyPermission "jaxp.debug", "read";

    // All JSPs need to be able to read this package
    permission java.lang.RuntimePermission "accessClassInPackage.org.apache.tomcat";

    // Precompiled JSPs need access to these packages.
    permission java.lang.RuntimePermission "accessClassInPackage.org.apache.jasper.el";
    permission java.lang.RuntimePermission "accessClassInPackage.org.apache.jasper.runtime";
    permission java.lang.RuntimePermission
     "accessClassInPackage.org.apache.jasper.runtime.*";

    // Precompiled JSPs need access to these system properties.
    permission java.util.PropertyPermission
     "org.apache.jasper.runtime.BodyContentImpl.LIMIT_BUFFER", "read";
    permission java.util.PropertyPermission
```

```
     "org.apache.el.parser.COERCE_TO_ZERO", "read";

    // The cookie code needs these.
    permission java.util.PropertyPermission
     "org.apache.catalina.STRICT_SERVLET_COMPLIANCE", "read";
    permission java.util.PropertyPermission
     "org.apache.tomcat.util.http.ServerCookie.STRICT_NAMING", "read";
    permission java.util.PropertyPermission
     "org.apache.tomcat.util.http.ServerCookie.FWD_SLASH_IS_SEPARATOR", "read";

    // Applications using Comet need to be able to access this package
    permission java.lang.RuntimePermission "accessClassInPackage.org.apache.catalina.comet";

    // Applications using WebSocket need to be able to access these packages
    permission java.lang.RuntimePermission "accessClassInPackage.org.apache.tomcat.websocket";
    permission java.lang.RuntimePermission "accessClassInPackage.org.apache.tomcat.websocket.server";
};


// The Manager application needs access to the following packages to support the
// session display functionality. These settings support the following
// configurations:
// - default CATALINA_HOME == CATALINA_BASE
// - CATALINA_HOME != CATALINA_BASE, per instance Manager in CATALINA_BASE
// - CATALINA_HOME != CATALINA_BASE, shared Manager in CATALINA_HOME
grant codeBase "file:${catalina.base}/webapps/manager/-" {
    permission java.lang.RuntimePermission "accessClassInPackage.org.apache.catalina";
    permission java.lang.RuntimePermission "accessClassInPackage.org.apache.catalina.ha.session";
    permission java.lang.RuntimePermission "accessClassInPackage.org.apache.catalina.manager";
    permission java.lang.RuntimePermission "accessClassInPackage.org.apache.catalina.manager.util";
    permission java.lang.RuntimePermission "accessClassInPackage.org.apache.catalina.util";
};
grant codeBase "file:${catalina.home}/webapps/manager/-" {
    permission java.lang.RuntimePermission "accessClassInPackage.org.apache.catalina";
```

```
    permission java.lang.RuntimePermission "accessClassInPackage.org.apache.catalina.ha.session";
    permission java.lang.RuntimePermission "accessClassInPackage.org.apache.catalina.manager";
    permission java.lang.RuntimePermission "accessClassInPackage.org.apache.catalina.manager.util";
    permission java.lang.RuntimePermission "accessClassInPackage.org.apache.catalina.util";
};

// You can assign additional permissions to particular web applications by
// adding additional "grant" entries here, based on the code base for that
// application, /WEB-INF/classes/, or /WEB-INF/lib/ jar files.
//
// Different permissions can be granted to JSP pages, classes loaded from
// the /WEB-INF/classes/ directory, all jar files in the /WEB-INF/lib/
// directory, or even to individual jar files in the /WEB-INF/lib/ directory.
//
// For instance, assume that the standard "examples" application
// included a JDBC driver that needed to establish a network connection to the
// corresponding database and used the scrape taglib to get the weather from
// the NOAA web server.  You might create a "grant" entries like this:
//
// The permissions granted to the context root directory apply to JSP pages.
// grant codeBase "file:${catalina.base}/webapps/examples/-" {
//      permission java.net.SocketPermission "dbhost.mycompany.com:5432", "connect";
//      permission java.net.SocketPermission "*.noaa.gov:80", "connect";
// };
//
// The permissions granted to the context WEB-INF/classes directory
// grant codeBase "file:${catalina.base}/webapps/examples/WEB-INF/classes/-" {
// };
//
// The permission granted to your JDBC driver
// grant codeBase "jar:file:${catalina.base}/webapps/examples/WEB-INF/lib/driver.jar!/-" {
//      permission java.net.SocketPermission "dbhost.mycompany.com:5432", "connect";
// };
// The permission granted to the scrape taglib
```

```
// grant codeBase "jar:file:${catalina.base}/webapps/examples/WEB-INF/lib/scrape.jar!/-" {
//      permission java.net.SocketPermission "*.noaa.gov:80", "connect";
// };
```

SER306 - Journalisation, Supervision et Clustering

Contenu du Module

Configuration des journaux

java.util.logging

Par défaut, Tomcat utilise le framework **java.util.logging** pour produire ses journaux.

Ce système de journalisation utilise le fichier **$CATALINA_HOME/conf/logging.properties** :

```
[root@centos7 bin]# cat $CATALINA_HOME/conf/logging.properties
# Licensed to the Apache Software Foundation (ASF) under one or more
# contributor license agreements.  See the NOTICE file distributed with
# this work for additional information regarding copyright ownership.
# The ASF licenses this file to You under the Apache License, Version 2.0
# (the "License"); you may not use this file except in compliance with
# the License.  You may obtain a copy of the License at
#
#     http://www.apache.org/licenses/LICENSE-2.0
#
# Unless required by applicable law or agreed to in writing, software
# distributed under the License is distributed on an "AS IS" BASIS,
# WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
# See the License for the specific language governing permissions and
# limitations under the License.

handlers = 1catalina.org.apache.juli.AsyncFileHandler, 2localhost.org.apache.juli.AsyncFileHandler,
3manager.org.apache.juli.AsyncFileHandler, 4host-manager.org.apache.juli.AsyncFileHandler,
java.util.logging.ConsoleHandler

.handlers = 1catalina.org.apache.juli.AsyncFileHandler, java.util.logging.ConsoleHandler

############################################################
# Handler specific properties.
# Describes specific configuration info for Handlers.
############################################################
```

```
1catalina.org.apache.juli.AsyncFileHandler.level = FINE
1catalina.org.apache.juli.AsyncFileHandler.directory = ${catalina.base}/logs
1catalina.org.apache.juli.AsyncFileHandler.prefix = catalina.

2localhost.org.apache.juli.AsyncFileHandler.level = FINE
2localhost.org.apache.juli.AsyncFileHandler.directory = ${catalina.base}/logs
2localhost.org.apache.juli.AsyncFileHandler.prefix = localhost.

3manager.org.apache.juli.AsyncFileHandler.level = FINE
3manager.org.apache.juli.AsyncFileHandler.directory = ${catalina.base}/logs
3manager.org.apache.juli.AsyncFileHandler.prefix = manager.

4host-manager.org.apache.juli.AsyncFileHandler.level = FINE
4host-manager.org.apache.juli.AsyncFileHandler.directory = ${catalina.base}/logs
4host-manager.org.apache.juli.AsyncFileHandler.prefix = host-manager.

java.util.logging.ConsoleHandler.level = FINE
java.util.logging.ConsoleHandler.formatter = org.apache.juli.OneLineFormatter


############################################################
# Facility specific properties.
# Provides extra control for each logger.
############################################################

org.apache.catalina.core.ContainerBase.[Catalina].[localhost].level = INFO
org.apache.catalina.core.ContainerBase.[Catalina].[localhost].handlers =
2localhost.org.apache.juli.AsyncFileHandler

org.apache.catalina.core.ContainerBase.[Catalina].[localhost].[/manager].level = INFO
org.apache.catalina.core.ContainerBase.[Catalina].[localhost].[/manager].handlers =
3manager.org.apache.juli.AsyncFileHandler

org.apache.catalina.core.ContainerBase.[Catalina].[localhost].[/host-manager].level = INFO
```

```
org.apache.catalina.core.ContainerBase.[Catalina].[localhost].[/host-manager].handlers = 4host-
manager.org.apache.juli.AsyncFileHandler

# For example, set the org.apache.catalina.util.LifecycleBase logger to log
# each component that extends LifecycleBase changing state:
#org.apache.catalina.util.LifecycleBase.level = FINE

# To see debug messages in TldLocationsCache, uncomment the following line:
#org.apache.jasper.compiler.TldLocationsCache.level = FINE
```

Dans ce fichier on constate la directive **handlers** :

```
...
handlers = 1catalina.org.apache.juli.AsyncFileHandler, 2localhost.org.apache.juli.AsyncFileHandler,
3manager.org.apache.juli.AsyncFileHandler, 4host-manager.org.apache.juli.AsyncFileHandler,
java.util.logging.ConsoleHandler
...
```

Il existe deux types de **handlers** :

- **org.apache.juli.AsyncFileHandler** qui écrit dans un fichier texte,
- **java.util.logging.ConsoleHandler** qui écrit sur la sortie standard.

Dans la déclaration des handlers, il faut spécifier un nom. Dans la déclaration ci-dessus, les noms sont :

- 1catalina,
- 2localhost,
- 3manager,
- 4host-manager.

La directive suivante permet de référencer le gestionnaire principal pour le serveur lui-même :

```
...
.handlers = 1catalina.org.apache.juli.AsyncFileHandler, java.util.logging.ConsoleHandler
```

```
...
```

Chaque handler doit ensuite être configuré :

```
1catalina.org.apache.juli.AsyncFileHandler.level = FINE
1catalina.org.apache.juli.AsyncFileHandler.directory = ${catalina.base}/logs
1catalina.org.apache.juli.AsyncFileHandler.prefix = catalina.

2localhost.org.apache.juli.AsyncFileHandler.level = FINE
2localhost.org.apache.juli.AsyncFileHandler.directory = ${catalina.base}/logs
2localhost.org.apache.juli.AsyncFileHandler.prefix = localhost.

3manager.org.apache.juli.AsyncFileHandler.level = FINE
3manager.org.apache.juli.AsyncFileHandler.directory = ${catalina.base}/logs
3manager.org.apache.juli.AsyncFileHandler.prefix = manager.

4host-manager.org.apache.juli.AsyncFileHandler.level = FINE
4host-manager.org.apache.juli.AsyncFileHandler.directory = ${catalina.base}/logs
4host-manager.org.apache.juli.AsyncFileHandler.prefix = host-manager.

java.util.logging.ConsoleHandler.level = FINE
java.util.logging.ConsoleHandler.formatter = org.apache.juli.OneLineFormatter
```

Les attributs communs à la classe org.apache.juli.AsyncFileHandler et java.util.logging.ConsoleHandler sont les suivants :

Attribut	**Description**
level	Spécifie le niveau de journalisation. Les niveaux sont SEVERE, CONFIG, INFO, WARN, FINE, FINEST ou ALL
formatter	Spécifie la classe utilisée pour formater le journal soit par défaut **java.util.logging.SimpleFormatter** soit **java.util.logging.XMLFormatter** pour générer une sortie au format XML

Les attributs spécifiques à la classe org.apache.juli.AsyncFileHandler sont les suivants :

Attribut	**Description**
prefix	Spécifie le nom du fichier

Attribut	Description
suffix	Spécifie l'extension du fichier
directory	Spécifie le répertoire de stockage des journaux

La rotation des journaux est journalier à 00h00. Le nom du journal aura don la forme suivante : **nom.AAAA.MM.JJ**.

La section suivante du fichier fournit un niveau de contrôle supplémentaire :

```
org.apache.catalina.core.ContainerBase.[Catalina].[localhost].level = INFO
org.apache.catalina.core.ContainerBase.[Catalina].[localhost].handlers =
2localhost.org.apache.juli.AsyncFileHandler

org.apache.catalina.core.ContainerBase.[Catalina].[localhost].[/manager].level = INFO
org.apache.catalina.core.ContainerBase.[Catalina].[localhost].[/manager].handlers =
3manager.org.apache.juli.AsyncFileHandler

org.apache.catalina.core.ContainerBase.[Catalina].[localhost].[/host-manager].level = INFO
org.apache.catalina.core.ContainerBase.[Catalina].[localhost].[/host-manager].handlers = 4host-
manager.org.apache.juli.AsyncFileHandler
```

Par exemple :

- [Catalina].[localhost] fait référence au hôte localhost,
- [Catalina].[localhost].[/manager] fait référence à l'application manager.

Il est à noter que les configurations spécifiques aux applications peuvent être incluses soit dans le fichier **$CATALINA_HOME/conf/logger.properties** soit dans un fichier logger.properties qui se trouve dans le répertoire **WEB-INF/classes** de l'application concernée.

Important : Pour mettre en place le debugging, utilisez le niveau **FINEST** ou **ALL**.

log4j

Il existe une alternative au framework **java.util.logging**, appelé **log4j**.

Téléchargez **log4j-1.2.17.zip** :

```
[root@centos7 ~]# wget http://apache.mirrors.ovh.net/ftp.apache.org/dist/logging/log4j/1.2.17/log4j-1.2.17.zip
```

Décompressez le fichier zip téléchargé :

```
[root@centos7 ~]# unzip log4j-1.2.17.zip
```

Copiez ensuite le fichier **log4j-1.2.17.jar** du répertoire **/root/apache-log4j-1.2.17/log4j-1.2.17.jar** vers le répertoire **$CATALINA_HOME/lib** en le renommant en **log4j.jar** :

```
[root@centos7 ~]# cp apache-log4j-1.2.17/log4j-1.2.17.jar $CATALINA_HOME/lib/log4j.jar
```

Téléchargez ensuite les fichiers **tomcat-juli-adapters.jar** et **tomcat-juli.jar** du dépôt **extras** d'Apache :

```
[root@centos7 ~]# wget
https://archive.apache.org/dist/tomcat/tomcat-8/v8.0.36/bin/extras/tomcat-juli-adapters.jar
[root@centos7 ~]# wget https://archive.apache.org/dist/tomcat/tomcat-8/v8.0.36/bin/extras/tomcat-juli.jar
```

Copiez le fichier **tomcat-juli-adapters.jar** ver **$CATALINA_HOME/lib/** :

```
[root@centos7 ~]# cp tomcat-juli-adapters.jar $CATALINA_HOME/lib/
```

Remplacez le fichier **$CATALINA_HOME/bin/tomcat-juli.jar** par le fichier téléchargé du même nom **/root/tomcat-juli.jar** :

```
[root@centos7 ~]# cp tomcat-juli.jar $CATALINA_HOME/bin/
cp: overwrite ‘/usr/tomcat8/bin/tomcat-juli.jar’? o
```

Créez maintenant le fichier **$CATALINA_HOME/lib/log4j.properties** qui configure log4j de façon à ce que celui-ci produise dans un premier temps les

mêmes journaux aux mêmes formats que le framework **java.util.logging** :

```
[root@centos7 ~]# vi $CATALINA_HOME/lib/log4j.properties
[root@centos7 ~]# cat $CATALINA_HOME/lib/log4j.properties
log4j.rootLogger = INFO, CATALINA

# Define all the appenders
log4j.appender.CATALINA = org.apache.log4j.DailyRollingFileAppender
log4j.appender.CATALINA.File = ${catalina.base}/logs/catalina
log4j.appender.CATALINA.Append = true
log4j.appender.CATALINA.Encoding = UTF-8
# Roll-over the log once per day
log4j.appender.CATALINA.DatePattern = '.'yyyy-MM-dd'.log'
log4j.appender.CATALINA.layout = org.apache.log4j.PatternLayout
log4j.appender.CATALINA.layout.ConversionPattern = %d [%t] %-5p %c- %m%n

log4j.appender.LOCALHOST = org.apache.log4j.DailyRollingFileAppender
log4j.appender.LOCALHOST.File = ${catalina.base}/logs/localhost
log4j.appender.LOCALHOST.Append = true
log4j.appender.LOCALHOST.Encoding = UTF-8
log4j.appender.LOCALHOST.DatePattern = '.'yyyy-MM-dd'.log'
log4j.appender.LOCALHOST.layout = org.apache.log4j.PatternLayout
log4j.appender.LOCALHOST.layout.ConversionPattern = %d [%t] %-5p %c- %m%n

log4j.appender.MANAGER = org.apache.log4j.DailyRollingFileAppender
log4j.appender.MANAGER.File = ${catalina.base}/logs/manager
log4j.appender.MANAGER.Append = true
log4j.appender.MANAGER.Encoding = UTF-8
log4j.appender.MANAGER.DatePattern = '.'yyyy-MM-dd'.log'
log4j.appender.MANAGER.layout = org.apache.log4j.PatternLayout
log4j.appender.MANAGER.layout.ConversionPattern = %d [%t] %-5p %c- %m%n

log4j.appender.HOST-MANAGER = org.apache.log4j.DailyRollingFileAppender
log4j.appender.HOST-MANAGER.File = ${catalina.base}/logs/host-manager
```

```
log4j.appender.HOST-MANAGER.Append = true
log4j.appender.HOST-MANAGER.Encoding = UTF-8
log4j.appender.HOST-MANAGER.DatePattern = '.'yyyy-MM-dd'.log'
log4j.appender.HOST-MANAGER.layout = org.apache.log4j.PatternLayout
log4j.appender.HOST-MANAGER.layout.ConversionPattern = %d [%t] %-5p %c- %m%n

log4j.appender.CONSOLE = org.apache.log4j.ConsoleAppender
log4j.appender.CONSOLE.Encoding = UTF-8
log4j.appender.CONSOLE.layout = org.apache.log4j.PatternLayout
log4j.appender.CONSOLE.layout.ConversionPattern = %d [%t] %-5p %c- %m%n

# Configure which loggers log to which appenders
log4j.logger.org.apache.catalina.core.ContainerBase.[Catalina].[localhost] = INFO, LOCALHOST
log4j.logger.org.apache.catalina.core.ContainerBase.[Catalina].[localhost].[/manager] =\
  INFO, MANAGER
log4j.logger.org.apache.catalina.core.ContainerBase.[Catalina].[localhost].[/host-manager] =\
  INFO, HOST-MANAGER
```

Déplacez le fichier **$CATALINA_HOME/conf/logging.properties**, utilisé par le framework **java.util.logging** :

```
[root@centos7 ~]# mv $CATALINA_HOME/conf/logging.properties /root
```

Dernièrement, redémarrez le serveur Tomcat :

```
[root@centos7 bin]# ./shutdown.sh
Using CATALINA_BASE:   /usr/tomcat8
Using CATALINA_HOME:   /usr/tomcat8
Using CATALINA_TMPDIR: /usr/tomcat8/temp
Using JRE_HOME:        /usr/lib/jvm/jre-1.8.0-openjdk-1.8.0.232.b09-0.el7_7.x86_64
Using CLASSPATH:       /usr/tomcat8/bin/bootstrap.jar:/usr/tomcat8/bin/tomcat-juli.jar
[root@centos7 bin]# ./startup.sh
Using CATALINA_BASE:   /usr/tomcat8
Using CATALINA_HOME:   /usr/tomcat8
Using CATALINA_TMPDIR: /usr/tomcat8/temp
```

```
Using JRE_HOME:        /usr/lib/jvm/jre-1.8.0-openjdk-1.8.0.232.b09-0.el7_7.x86_64
Using CLASSPATH:       /usr/tomcat8/bin/bootstrap.jar:/usr/tomcat8/bin/tomcat-juli.jar
Tomcat started.
```

Pour plus d'information concernant la journalisation, consultez cette **page**.

Supervision

JMeter

Ouvrez un terminal dans l'interface graphique de votre VM.

La gestion de la montée en charge de Tomcat peut être faite avec le produit libre **JMeter**. Pour l'obtenir, il convient de le télécharger :

```
[root@centos7 ~]# wget https://archive.apache.org/dist/jmeter/binaries/apache-jmeter-3.0.tgz
```

Décompressez l'archive dans **$CATALINA_HOME/JMeter** :

```
[root@centos7 ~]# mkdir $CATALINA_HOME/JMeter
[root@centos7 ~]# mv apache-jmeter-3.0.tgz $CATALINA_HOME/JMeter
[root@centos7 ~]# cd $CATALINA_HOME/JMeter
[root@centos7 JMeter]# tar xvf apache-jmeter-3.0.tgz
```

L'arborescence obtenu est :

```
[root@centos7 JMeter]# ls
apache-jmeter-3.0  apache-jmeter-3.0.tgz
[root@centos7 JMeter]# cd apache-jmeter-3.0/
[root@centos7 apache-jmeter-3.0]# ls
bin  docs  extras  lib  LICENSE  licenses  NOTICE  printable_docs  README
```

Définissez maintenant la variable **$JMETER_HOME** :

```
[root@centos7 apache-jmeter-3.0]# JMETER_HOME=/usr/tomcat8/JMeter/apache-jmeter-3.0
[root@centos7 apache-jmeter-3.0]# export JMETER_HOME
[root@centos7 apache-jmeter-3.0]# echo $JMETER_HOME
/usr/tomcat8/JMeter/apache-jmeter-3.0
```

Lancez ensuite JMeter dans un terminal de votre VM en mode graphique :

```
[root@redhat JMeter]# /usr/tomcat8/JMeter/apache-jmeter-3.0/bin/jmeter
```

Testez cet outil en utilisant les différents fichiers mis-à-disposition lors de l'installation de JMeter :

```
[root@centos7 apache-jmeter-3.0]# updatedb
[root@centos7 apache-jmeter-3.0]# locate .jmx
/usr/tomcat8/JMeter/apache-jmeter-3.0/bin/examples/CSVSample.jmx
/usr/tomcat8/JMeter/apache-jmeter-3.0/bin/examples/PerformanceTestPlanMemoryThread.jmx
/usr/tomcat8/JMeter/apache-jmeter-3.0/bin/templates/BeanShellSampler.jmx
/usr/tomcat8/JMeter/apache-jmeter-3.0/bin/templates/build-adv-web-test-plan.jmx
/usr/tomcat8/JMeter/apache-jmeter-3.0/bin/templates/build-ftp-test-plan.jmx
/usr/tomcat8/JMeter/apache-jmeter-3.0/bin/templates/build-ldap-ext-test-plan.jmx
/usr/tomcat8/JMeter/apache-jmeter-3.0/bin/templates/build-ldap-test-plan.jmx
/usr/tomcat8/JMeter/apache-jmeter-3.0/bin/templates/build-web-test-plan.jmx
/usr/tomcat8/JMeter/apache-jmeter-3.0/bin/templates/build-webservice-test-plan.jmx
/usr/tomcat8/JMeter/apache-jmeter-3.0/bin/templates/jdbc.jmx
/usr/tomcat8/JMeter/apache-jmeter-3.0/bin/templates/mongodb.jmx
/usr/tomcat8/JMeter/apache-jmeter-3.0/bin/templates/recording-with-think-time.jmx
/usr/tomcat8/JMeter/apache-jmeter-3.0/bin/templates/recording.jmx
/usr/tomcat8/JMeter/apache-jmeter-3.0/extras/Test.jmx
/usr/tomcat8/JMeter/apache-jmeter-3.0/printable_docs/demos/AssertionTestPlan.jmx
/usr/tomcat8/JMeter/apache-jmeter-3.0/printable_docs/demos/AuthManagerTestPlan.jmx
/usr/tomcat8/JMeter/apache-jmeter-3.0/printable_docs/demos/ForEachTest2.jmx
/usr/tomcat8/JMeter/apache-jmeter-3.0/printable_docs/demos/HeaderManagerTestPlan.jmx
/usr/tomcat8/JMeter/apache-jmeter-3.0/printable_docs/demos/InterleaveTestPlan.jmx
```

```
/usr/tomcat8/JMeter/apache-jmeter-3.0/printable_docs/demos/InterleaveTestPlan2.jmx
/usr/tomcat8/JMeter/apache-jmeter-3.0/printable_docs/demos/JDBC-Pre-Post-Processor.jmx
/usr/tomcat8/JMeter/apache-jmeter-3.0/printable_docs/demos/JMSPointToPoint.jmx
/usr/tomcat8/JMeter/apache-jmeter-3.0/printable_docs/demos/LoopTestPlan.jmx
/usr/tomcat8/JMeter/apache-jmeter-3.0/printable_docs/demos/OnceOnlyTestPlan.jmx
/usr/tomcat8/JMeter/apache-jmeter-3.0/printable_docs/demos/ProxyServerTestPlan.jmx
/usr/tomcat8/JMeter/apache-jmeter-3.0/printable_docs/demos/RegEx-User-Parameters.jmx
/usr/tomcat8/JMeter/apache-jmeter-3.0/printable_docs/demos/SimpleTestPlan.jmx
/usr/tomcat8/JMeter/apache-jmeter-3.0/printable_docs/demos/URLRewritingExample.jmx
/usr/tomcat8/JMeter/apache-jmeter-3.0/printable_docs/demos/forEachTestPlan.jmx
```

Important : Pour plus d'information concernant cet outil, consultez la page **http://jmeter.apache.org/changes.html**.

Interface JMX

L'interface JMX est un outil complémentaire à JMeter car il est capable de montrer :

- le comportement interne du serveur,
- le nombre de connexions JDBC disponibles à un instant t,
- le nombre de threads occupés dans un connecteur.

JMX ou *Java Management eXtensions* est un API Java qui extrait des informations des **MBeans**. Les MBeans sont créés dans le fichier $CATALINA_HOME/conf/server.xml en utilisant des éléments <Listener> :

```
...
  <Listener className="org.apache.catalina.startup.VersionLoggerListener" />
  <Listener className="org.apache.catalina.security.SecurityListener" />
  <Listener className="org.apache.catalina.core.AprLifecycleListener" SSLEngine="on" />
  <Listener className="org.apache.catalina.core.JreMemoryLeakPreventionListener" />
  <Listener className="org.apache.catalina.mbeans.GlobalResourcesLifecycleListener" />
```

```
  <Listener className="org.apache.catalina.core.ThreadLocalLeakPreventionListener" />
...
```

Revenez à l'authentification en utilisant le fichier **tomcat-users.xml** en éditant le fichier **$CATALINA_HOME/conf/server.xml** :

```
[root@centos7 bin]# vi $CATALINA_HOME/conf/server.xml
[root@centos7 bin]# cat $CATALINA_HOME/conf/server.xml
...
        <Realm className="org.apache.catalina.realm.UserDatabaseRealm"
               resourceName="UserDatabase" digest="sha" />
...
      <!-- <Realm  className="org.apache.catalina.realm.JNDIRealm"
          connectionURL="ldap://localhost:389"
         connectionName="cn=Manager,o=fenestros.loc"
     connectionPassword="fenestros"
               roleBase="ou=roles,o=fenestros.loc"
               roleName="cn"
             roleSearch="(uniqueMember={0})"
           userPassword="userPassword"
            userPattern="cn={0},ou=utilisateurs,o=fenestros.loc" /> -->
...
```

Éditez ensuite le fichier **$CATALINA_HOME/conf/tomcat-users.xml** :

```
[root@centos7 bin]# vi $CATALINA_HOME/conf/tomcat-users.xml
[root@centos7 bin]# cat $CATALINA_HOME/conf/tomcat-users.xml
<?xml version='1.0' encoding='utf-8'?>
<tomcat-users xmlns="http://tomcat.apache.org/xml"
              xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
              xsi:schemaLocation="http://tomcat.apache.org/xml tomcat-users.xsd"
              version="1.0">
  <role rolename="tomcat"/>
  <role rolename="role1"/>
  <role rolename="manager-script"/>
```

```
  <role rolename="manager-gui"/>
  <role rolename="manager-jmx"/>
  <user username="tomcat" password="tomcat" roles="tomcat"/>
  <user username="both" password="tomcat" roles="tomcat,role1"/>
  <user username="role1" password="tomcat" roles="role1"/>
  <user username="admin"
password="75affcc6adf7ec7cd21f6479b8fb87b89a550b2602e613715d57d862b15c7c17$1$015b4b320315c87572918dbcc08b65a240d0
a750" roles="manager-script,manager-gui,manager-jmx"/>
</tomcat-users>
```

Redémarrez le serveur Tomcat :

```
[root@centos7 apache-jmeter-3.0]# cd $CATALINA_HOME/bin
[root@centos7 bin]# ./shutdown.sh
Using CATALINA_BASE:   /usr/tomcat8
Using CATALINA_HOME:   /usr/tomcat8
Using CATALINA_TMPDIR: /usr/tomcat8/temp
Using JRE_HOME:        /usr/lib/jvm/jre-1.8.0-openjdk-1.8.0.232.b09-0.el7_7.x86_64
Using CLASSPATH:       /usr/tomcat8/bin/bootstrap.jar:/usr/tomcat8/bin/tomcat-juli.jar
[root@centos7 bin]# ./startup.sh
Using CATALINA_BASE:   /usr/tomcat8
Using CATALINA_HOME:   /usr/tomcat8
Using CATALINA_TMPDIR: /usr/tomcat8/temp
Using JRE_HOME:        /usr/lib/jvm/jre-1.8.0-openjdk-1.8.0.232.b09-0.el7_7.x86_64
Using CLASSPATH:       /usr/tomcat8/bin/bootstrap.jar:/usr/tomcat8/bin/tomcat-juli.jar
Tomcat started.
```

Testez que l'authentification fonctionne correctement :

```
[root@centos7 bin]# lynx --dump -auth admin:fenestros "http://www.i2tch.loc:8080/manager/text/serverinfo"
OK - Server info
Tomcat Version: Apache Tomcat/8.0.36
OS Name: Linux
OS Version: 3.10.0-1062.4.1.el7.x86_64
```

```
OS Architecture: amd64
JVM Version: 1.8.0_232-b09
JVM Vendor: Oracle Corporation
```

L'application **manager** propose un client JMX sous la forme d'un proxy JMX, **jmxproxy**.

Saisissez donc la commande suivante :

```
[root@centos7 bin]# lynx --dump -auth admin:fenestros "http://www.i2tch.loc:8080/manager/jmxproxy/?qry=*:*" |
more
OK - Number of results: 174

Name: Catalina:type=Service
modelerType: org.apache.catalina.mbeans.ServiceMBean
stateName: STARTED
connectorNames: Array[javax.management.ObjectName] of length 3
        Catalina:type=Connector,port=8080
        Catalina:type=Connector,port=8443
        Catalina:type=Connector,port=8009
name: Catalina
managedResource: StandardService[Catalina]

Name: Catalina:j2eeType=Servlet,WebModule=//localhost/examples,name=stock,J2EEAp
plication=none,J2EEServer=none
modelerType: org.apache.catalina.mbeans.ContainerMBean
maxTime: 0
requestCount: 0
stateManageable: false
servletClass: async.AsyncStockServlet
countAllocated: 0
available: 0
backgroundProcessorDelay: -1
processingTime: 0
loadOnStartup: -1
```

```
singleThreadModel: false
loadTime: 2
stateName: STARTED
minTime: 9223372036854775807
classLoadTime: 2
--More--
```

Saisissez maintenant la commande suivante :

```
[root@centos7 bin]# lynx --dump -auth admin:fenestros
"http://www.i2tch.loc:8080/manager/jmxproxy/?qry=Catalina:type=Connector,*" | more
OK - Number of results: 3

Name: Catalina:type=Connector,port=8009
modelerType: org.apache.catalina.mbeans.ConnectorMBean
maxPostSize: 2097152
scheme: http
acceptCount: 100
secure: false
threadPriority: 5
ajpFlush: true
maxSavePostSize: 4096
proxyPort: 0
protocol: AJP/1.3
maxParameterCount: 10000
useIPVHosts: false
stateName: STARTED
redirectPort: 8443
allowTrace: false
protocolHandlerClassName: org.apache.coyote.ajp.AjpNioProtocol
maxThreads: 200
connectionTimeout: -1
tcpNoDelay: true
useBodyEncodingForURI: false
```

```
connectionLinger: -1
processorCache: 200
keepAliveTimeout: -1
localPort: 8009
enableLookups: false
packetSize: 8192
--More--
```

Saisissez maintenant la commande suivante :

```
[root@centos7 bin]# lynx --dump -auth admin:fenestros
"http://www.i2tch.loc:8080/manager/jmxproxy/?qry=Catalina:type=ThreadPool,*"
OK - Number of results: 3

Name: Catalina:type=ThreadPool,name="http-nio-8443"
modelerType: org.apache.tomcat.util.modeler.BaseModelMBean
currentThreadsBusy: 0
selectorTimeout: 1000
paused: false
socketProperties: org.apache.tomcat.util.net.SocketProperties@346f7a2a
useCometTimeout: true
currentThreadCount: 0
usePolling: true
maxThreads: 200
tcpNoDelay: true
algorithm: SunX509
maxKeepAliveRequests: 100
keepAliveTimeout: 60000
localPort: 8443
pollerThreadCount: 1
acceptorThreadCount: 1
soTimeout: 60000
daemon: true
minSpareThreads: 10
```

```
acceptorThreadPriority: 5
backlog: 100
maxHeaderCount: 100
port: 8443
keystoreType: JKS
name: http-nio-8443
soLinger: -1
sslProtocol: TLS
sessionTimeout: 86400
useComet: true
clientAuth: false
connectionCount: 1
threadPriority: 5
executorTerminationTimeoutMillis: 5000
running: true
sslEnabledProtocolsArray: Array[java.lang.String] of length 0
ciphers: HIGH:!aNULL:!eNULL:!EXPORT:!DES:!RC4:!MD5:!kRSA
sSLEnabled: false
selectorPool: org.apache.tomcat.util.net.NioSelectorPool@8fecc2b
maxConnections: 10000
keepAliveCount: 0
deferAccept: false
useSendfile: true
oomParachute: 1048576
bindOnInit: true
pollerThreadPriority: 5
keystoreFile: /root/.keystore
useServerCipherSuitesOrder:

Name: Catalina:type=ThreadPool,name="ajp-nio-8009"
modelerType: org.apache.tomcat.util.modeler.BaseModelMBean
currentThreadsBusy: 0
selectorTimeout: 1000
paused: false
```

```
socketProperties: org.apache.tomcat.util.net.SocketProperties@50b03315
useCometTimeout: true
currentThreadCount: 0
usePolling: true
maxThreads: 200
tcpNoDelay: true
algorithm: SunX509
maxKeepAliveRequests: 100
keepAliveTimeout: -1
localPort: 8009
pollerThreadCount: 1
acceptorThreadCount: 1
soTimeout: -1
daemon: true
minSpareThreads: 10
acceptorThreadPriority: 5
backlog: 100
maxHeaderCount: 100
port: 8009
keystoreType: JKS
name: ajp-nio-8009
soLinger: -1
sslProtocol: TLS
sessionTimeout: 86400
useComet: true
clientAuth: false
connectionCount: 1
threadPriority: 5
executorTerminationTimeoutMillis: 5000
running: true
sslEnabledProtocolsArray: Array[java.lang.String] of length 0
ciphers: HIGH:!aNULL:!eNULL:!EXPORT:!DES:!RC4:!MD5:!kRSA
sSLEnabled: false
selectorPool: org.apache.tomcat.util.net.NioSelectorPool@3b104969
```

```
maxConnections: 10000
keepAliveCount: 0
deferAccept: false
useSendfile: false
oomParachute: 1048576
bindOnInit: true
pollerThreadPriority: 5
keystoreFile: /root/.keystore
useServerCipherSuitesOrder:

Name: Catalina:type=ThreadPool,name="http-nio-8080"
modelerType: org.apache.tomcat.util.modeler.BaseModelMBean
currentThreadsBusy: 2
selectorTimeout: 1000
paused: false
socketProperties: org.apache.tomcat.util.net.SocketProperties@182321ae
useCometTimeout: true
currentThreadCount: 8
usePolling: true
maxThreads: 200
tcpNoDelay: true
algorithm: SunX509
maxKeepAliveRequests: 100
keepAliveTimeout: 20000
localPort: 8080
pollerThreadCount: 1
acceptorThreadCount: 1
soTimeout: 20000
daemon: true
minSpareThreads: 10
acceptorThreadPriority: 5
backlog: 100
maxHeaderCount: 100
port: 8080
```

```
keystoreType: JKS
name: http-nio-8080
soLinger: -1
sslProtocol: TLS
sessionTimeout: 86400
useComet: true
clientAuth: false
connectionCount: 3
threadPriority: 5
executorTerminationTimeoutMillis: 5000
running: true
sslEnabledProtocolsArray: Array[java.lang.String] of length 0
ciphers: HIGH:!aNULL:!eNULL:!EXPORT:!DES:!RC4:!MD5:!kRSA
sSLEnabled: false
selectorPool: org.apache.tomcat.util.net.NioSelectorPool@596e32c2
maxConnections: 10000
keepAliveCount: 1
deferAccept: false
useSendfile: true
oomParachute: 1048576
bindOnInit: true
pollerThreadPriority: 5
keystoreFile: /root/.keystore
useServerCipherSuitesOrder:
```

Notez la valeur de **maxThreads** dans la section **name="http-nio-8080"** :

```
...
maxThreads: 200
...
```

Pour modifier la valeur de maxThreads, il faut créer le fichier $CATALINA_HOME/bin/setenv.sh :

```
[root@centos7 bin]# vi setenv.sh
```

```
[root@centos7 bin]# cat setenv.sh
export JAVA_OPTS="-Dcom.sun.management.jmxremote=true -Dcom.sun.management.jmxremote.ssl=false -
Dcom.sun.management.jmxremote.authenticate=false"
[root@centos7 bin]# chmod ugo+x setenv.sh
[root@centos7 bin]# ls -l setenv.sh
-rwxr-xr-x 1 root root 208 Jun 28 23:40 setenv.sh
```

Redémarrez le serveur Tomcat pour une prise en compte du fichier setenv.sh :

```
[root@centos7 bin]# ./shutdown.sh
Using CATALINA_BASE:   /usr/tomcat8
Using CATALINA_HOME:   /usr/tomcat8
Using CATALINA_TMPDIR: /usr/tomcat8/temp
Using JRE_HOME:        /usr/lib/jvm/jre-1.8.0-openjdk-1.8.0.232.b09-0.el7_7.x86_64
Using CLASSPATH:       /usr/tomcat8/bin/bootstrap.jar:/usr/tomcat8/bin/tomcat-juli.jar
[root@centos7 bin]# ./startup.sh
Using CATALINA_BASE:   /usr/tomcat8
Using CATALINA_HOME:   /usr/tomcat8
Using CATALINA_TMPDIR: /usr/tomcat8/temp
Using JRE_HOME:        /usr/lib/jvm/jre-1.8.0-openjdk-1.8.0.232.b09-0.el7_7.x86_64
Using CLASSPATH:       /usr/tomcat8/bin/bootstrap.jar:/usr/tomcat8/bin/tomcat-juli.jar
Tomcat started.
```

Maintenant saisissez l'URL suivant dans un navigateur en mode graphique pour modifier la valeur de MaxThreads :

```
http://www.i2tch.loc:8080/manager/jmxproxy/?set=Catalina:type=ThreadPool,name="http-nio-8080"&att=maxThreads&val=
300
```

Important : Le navigateur vous demandera de renseigner un utilisateur et un mot de passe : admin/fenestros.

Vous obtiendrez un résultat similaire à celui-ci :

```
OK - Attribute set
```

Saisissez la commande suivante pour vérifier la prise en compte de la modification :

```
[root@centos7 bin]# lynx --dump -auth admin:fenestros
"http://www.i2tch.loc:8080/manager/jmxproxy/?qry=Catalina:type=ThreadPool,*"
...
Name: Catalina:type=ThreadPool,name="http-nio-8080"
modelerType: org.apache.tomcat.util.modeler.BaseModelMBean
currentThreadsBusy: 1
selectorTimeout: 1000
paused: false
socketProperties: org.apache.tomcat.util.net.SocketProperties@37cfc160
useCometTimeout: true
currentThreadCount: 10
usePolling: true
maxThreads: 300
tcpNoDelay: true
algorithm: SunX509
maxKeepAliveRequests: 100
keepAliveTimeout: 20000
localPort: 8080
pollerThreadCount: 1
acceptorThreadCount: 1
soTimeout: 20000
daemon: true
minSpareThreads: 10
acceptorThreadPriority: 5
backlog: 100
maxHeaderCount: 100
port: 8080
keystoreType: JKS
name: http-nio-8080
soLinger: -1
```

```
sslProtocol: TLS
sessionTimeout: 86400
useComet: true
clientAuth: false
connectionCount: 2
threadPriority: 5
executorTerminationTimeoutMillis: 5000
running: true
sslEnabledProtocolsArray: Array[java.lang.String] of length 0
ciphers: HIGH:!aNULL:!eNULL:!EXPORT:!DES:!RC4:!MD5:!kRSA
sSLEnabled: false
selectorPool: org.apache.tomcat.util.net.NioSelectorPool@65151909
maxConnections: 10000
keepAliveCount: 1
deferAccept: false
useSendfile: true
oomParachute: 1048576
bindOnInit: true
pollerThreadPriority: 5
keystoreFile: /root/.keystore
useServerCipherSuitesOrder:
```

JConsole

Pour utiliser JConsole, commencez par créer le fichier des utilisateurs et des mots de passe :

```
[root@centos7 bin]# cd ../conf
[root@centos7 conf]# vi jmxremote.access
[root@centos7 conf]# cat jmxremote.access
administrator   readwrite
operator    readonly
[root@centos7 conf]# vi jmxremote.password
[root@centos7 conf]# cat jmxremote.password
```

```
administrator   fenestros
operator    tomcat
[root@centos7 conf]# chmod 600 jmxremote.password
```

Dans un premier temps vous allez mettre en place une connexion anonyme à JConsole. Éditez donc votre fichier **$CATALINA_HOME/bin/setenv.sh** :

```
[root@centos7 conf]# cd ../bin
[root@centos7 bin]# vi setenv.sh
[root@centos7 bin]# cat setenv.sh
export JAVA_OPTS="-Dcom.sun.management.jmxremote=true -Dcom.sun.management.jmxremote.ssl=false -
Dcom.sun.management.jmxremote.authenticate=false -Dcom.sun.management.jmxremote.port=9004"
```

Redémarrez maintenant le serveur Tomcat :

```
[root@centos7 bin]# ./shutdown.sh
Using CATALINA_BASE:   /usr/tomcat8
Using CATALINA_HOME:   /usr/tomcat8
Using CATALINA_TMPDIR: /usr/tomcat8/temp
Using JRE_HOME:        /usr/lib/jvm/jre-1.8.0-openjdk-1.8.0.232.b09-0.el7_7.x86_64
Using CLASSPATH:       /usr/tomcat8/bin/bootstrap.jar:/usr/tomcat8/bin/tomcat-juli.jar
[root@centos7 bin]# ./startup.sh
Using CATALINA_BASE:   /usr/tomcat8
Using CATALINA_HOME:   /usr/tomcat8
Using CATALINA_TMPDIR: /usr/tomcat8/temp
Using JRE_HOME:        /usr/lib/jvm/jre-1.8.0-openjdk-1.8.0.232.b09-0.el7_7.x86_64
Using CLASSPATH:       /usr/tomcat8/bin/bootstrap.jar:/usr/tomcat8/bin/tomcat-juli.jar
Tomcat started.
```

Lancez la commande **jconsole** dans un terminal de l'interface graphique de votre VM. Cochez **Remote Process** et utilisez l'adresse **localhost:9004 sans** stipuler un utilisateur et un mot de passe.

Pour mettre en place une autorisation en utilisant les fichiers **jmxremote.access** et **jmxremote.password**, il convient d'éditer de nouveau le fichier **$CATALINA_HOME/bin/setenv.sh** :

```
[root@centos7 bin]# vi setenv.sh
[root@centos7 bin]# cat setenv.sh
export JAVA_OPTS="-Dcom.sun.management.jmxremote=true -Dcom.sun.management.jmxremote.ssl=false -
Dcom.sun.management.jmxremote.port=9004 -
Dcom.sun.management.jmxremote.password.file=$CATALINA_HOME/conf/jmxremote.password -
Dcom.sun.management.jmxremote.access.file=$CATALINA_HOME/conf/jmxremote.access"
```

Redémarrez le serveur Tomcat :

```
[root@centos7 bin]# ./shutdown.sh
Using CATALINA_BASE:   /usr/tomcat8
Using CATALINA_HOME:   /usr/tomcat8
Using CATALINA_TMPDIR: /usr/tomcat8/temp
Using JRE_HOME:        /usr/lib/jvm/jre-1.8.0-openjdk-1.8.0.232.b09-0.el7_7.x86_64
Using CLASSPATH:       /usr/tomcat8/bin/bootstrap.jar:/usr/tomcat8/bin/tomcat-juli.jar
[root@centos7 bin]# ./startup.sh
Using CATALINA_BASE:   /usr/tomcat8
Using CATALINA_HOME:   /usr/tomcat8
Using CATALINA_TMPDIR: /usr/tomcat8/temp
Using JRE_HOME:        /usr/lib/jvm/jre-1.8.0-openjdk-1.8.0.232.b09-0.el7_7.x86_64
Using CLASSPATH:       /usr/tomcat8/bin/bootstrap.jar:/usr/tomcat8/bin/tomcat-juli.jar
Tomcat started.
```

Lancez la commande **jconsole** dans un terminal de l'interface graphique de votre VM. Cochez **Remote Process** et utilisez l'adresse **localhost:9004 en** stipulant un utilisateur et un mot de passe dans les fichiers **jmxremote.access** et **jmxremote.password** respectivement.

Clustering avec Tomcat

Préparation

Créez maintenant deux répertoires en dessous de $CATALINA_HOME :

```
[root@centos7 ~]# mkdir $CATALINA_HOME/tomcat1 $CATALINA_HOME/tomcat2
```

Arrêtez le serveur Tomcat et copiez les répertoires $CATALINA_HOME/conf, $CATALINA_HOME/logs, $CATALINA_HOME/temp, $CATALINA_HOME/webapps, $CATALINA_HOME/work dans les répertoires $CATALINA_HOME/tomcat1 et $CATALINA_HOME/tomcat2 :

```
[root@centos7 ~]# cd $CATALINA_HOME
[root@centos7 tomcat8]# cp -rp conf/ tomcat1/
[root@centos7 tomcat8]# cp -rp logs/ tomcat1
[root@centos7 tomcat8]# cp -rp temp/ tomcat1
[root@centos7 tomcat8]# cp -rp webapps/ tomcat1
[root@centos7 tomcat8]# cp -rp work/ tomcat1
[root@centos7 tomcat8]# cp -rp conf/ tomcat2/
[root@centos7 tomcat8]# cp -rp logs/ tomcat2/
[root@centos7 tomcat8]# cp -rp temp/ tomcat2/
[root@centos7 tomcat8]# cp -rp webapps/ tomcat2/
[root@centos7 tomcat8]# cp -rp work/ tomcat2/
```

Supprimez les répertoires $CATALINA_HOME/conf, $CATALINA_HOME/logs, $CATALINA_HOME/temp, $CATALINA_HOME/webapps, $CATALINA_HOME/work :

```
[root@centos7 tomcat8]# rm -rf conf/ logs/ temp/ webapps/ work/
```

Supprimez maintenant le fichier **$CATALINA_HOME/bin/setenv.sh** :

```
[root@centos7 tomcat8]# rm -rf bin/setenv.sh
```

Créez maintenant les scripts de démarrage et d'arrêt de chaque instance de Tomcat :

```
[root@centos7 tomcat8]# cd bin
[root@centos7 bin]# vi startTomcat1
[root@centos7 bin]# cat startTomcat1
#!/bin/bash
export CATALINA_BASE=/usr/tomcat8/tomcat1
```

```
. $CATALINA_HOME/bin/startup.sh

[root@centos7 bin]# vi stopTomcat1
[root@centos7 bin]# cat stopTomcat1
#!/bin/bash
export CATALINA_BASE=/usr/tomcat8/tomcat1
. $CATALINA_HOME/bin/shutdown.sh

[root@centos7 bin]# vi startTomcat2
[root@centos7 bin]# cat startTomcat2
export CATALINA_BASE=/usr/tomcat8/tomcat2
. $CATALINA_HOME/bin/startup.sh

[root@centos7 bin]# vi stopTomcat2
[root@centos7 bin]# cat stopTomcat2
#!/bin/bash
export CATALINA_BASE=/usr/tomcat8/tomcat2
. $CATALINA_HOME/bin/shutdown.sh
```

Rendez les scripts exécutables :

```
[root@centos7 bin]# chmod a+x startTomcat1
[root@centos7 bin]# chmod a+x startTomcat2
[root@centos7 bin]# chmod a+x stopTomcat1
[root@centos7 bin]# chmod a+x stopTomcat2
[root@centos7 bin]# ls -l | grep startT
-rwxr-xr-x  1 root root     86 Jun 29 01:47 startTomcat1
-rwxr-xr-x  1 root root     86 Jun 29 01:51 startTomcat2
[root@centos7 bin]# ls -l | grep stopT
-rwxr-xr-x  1 root root     87 Jun 29 01:50 stopTomcat1
-rwxr-xr-x  1 root root     87 Jun 29 01:52 stopTomcat2
```

Modifiez les ports dans le fichier server.xml de chaque installation de Tomcat en utilisant VI :

```
[root@centos7 bin]# vi /usr/tomcat8/tomcat1/conf/server.xml
[root@centos7 bin]# vi /usr/tomcat8/tomcat2/conf/server.xml
```

Les commandes VI suivantes peuvent vous aider :

```
Pour le fichier /usr/tomcat8/tomcat1/conf/server.xml :
:g/8080/s//8180/g
:g/8009/s//8109/g
:g/8005/s//8105/g
:g/8443/s//8143/g
Pour le fichier /usr/tomcat8/tomcat2/conf/server.xml :
:g/8080/s//8280/g
:g/8009/s//8209/g
:g/8005/s//8205/g
:g/8443/s//8243/g
```

Démarrez les deux instances de Tomcat :

```
[root@centos7 bin]# ./startTomcat1
Using CATALINA_BASE:   /usr/tomcat8/tomcat1
Using CATALINA_HOME:   /usr/tomcat8
Using CATALINA_TMPDIR: /usr/tomcat8/tomcat1/temp
Using JRE_HOME:        /usr/lib/jvm/jre-1.8.0-openjdk-1.8.0.232.b09-0.el7_7.x86_64
Using CLASSPATH:       /usr/tomcat8/bin/bootstrap.jar:/usr/tomcat8/bin/tomcat-juli.jar
Tomcat started.
[root@centos7 bin]# ps aux | grep tomcat
root     25696 30.0  4.4 2399312 67900 pts/0   Sl   02:47   0:04 /usr/lib/jvm/jre-1.8.0-
openjdk-1.8.0.232.b09-0.el7_7.x86_64/bin/java -
Djava.util.logging.config.file=/usr/tomcat8/tomcat1/conf/logging.properties -
Djava.util.logging.manager=org.apache.juli.ClassLoaderLogManager -Djdk.tls.ephemeralDHKeySize=2048 -
Djava.endorsed.dirs=/usr/tomcat8/endorsed -classpath /usr/tomcat8/bin/bootstrap.jar:/usr/tomcat8/bin/tomcat-
juli.jar -Dcatalina.base=/usr/tomcat8/tomcat1 -Dcatalina.home=/usr/tomcat8 -
Djava.io.tmpdir=/usr/tomcat8/tomcat1/temp org.apache.catalina.startup.Bootstrap start
root     25785  0.0  0.0 112644   968 pts/0    R+   02:47   0:00 grep --color=auto tomcat
```

```
[root@centos7 bin]# ./startTomcat2
Using CATALINA_BASE:   /usr/tomcat8/tomcat2
Using CATALINA_HOME:   /usr/tomcat8
Using CATALINA_TMPDIR: /usr/tomcat8/tomcat2/temp
Using JRE_HOME:        /usr/lib/jvm/jre-1.8.0-openjdk-1.8.0.232.b09-0.el7_7.x86_64
Using CLASSPATH:       /usr/tomcat8/bin/bootstrap.jar:/usr/tomcat8/bin/tomcat-juli.jar
Tomcat started.
[root@centos7 bin]# ps aux | grep tomcat
root     25696 32.1  5.2 2403492 80468 pts/0   Sl   02:47   0:07 /usr/lib/jvm/jre-1.8.0-
openjdk-1.8.0.232.b09-0.el7_7.x86_64/bin/java -
Djava.util.logging.config.file=/usr/tomcat8/tomcat1/conf/logging.properties -
Djava.util.logging.manager=org.apache.juli.ClassLoaderLogManager -Djdk.tls.ephemeralDHKeySize=2048 -
Djava.endorsed.dirs=/usr/tomcat8/endorsed -classpath /usr/tomcat8/bin/bootstrap.jar:/usr/tomcat8/bin/tomcat-
juli.jar -Dcatalina.base=/usr/tomcat8/tomcat1 -Dcatalina.home=/usr/tomcat8 -
Djava.io.tmpdir=/usr/tomcat8/tomcat1/temp org.apache.catalina.startup.Bootstrap start
root     25817 32.6  2.4 2381580 37172 pts/0   Sl   02:47   0:00 /usr/lib/jvm/jre-1.8.0-
openjdk-1.8.0.232.b09-0.el7_7.x86_64/bin/java -
Djava.util.logging.config.file=/usr/tomcat8/tomcat2/conf/logging.properties -
Djava.util.logging.manager=org.apache.juli.ClassLoaderLogManager -Djdk.tls.ephemeralDHKeySize=2048 -
Djava.endorsed.dirs=/usr/tomcat8/endorsed -classpath /usr/tomcat8/bin/bootstrap.jar:/usr/tomcat8/bin/tomcat-
juli.jar -Dcatalina.base=/usr/tomcat8/tomcat2 -Dcatalina.home=/usr/tomcat8 -
Djava.io.tmpdir=/usr/tomcat8/tomcat2/temp org.apache.catalina.startup.Bootstrap start
root     25843  0.0  0.0 112644   968 pts/0    S+   02:47   0:00 grep --color=auto tomcat
```

Vérifiez maintenant que les deux instances peuvent être arrêtés :

```
[root@centos7 bin]# ./stopTomcat2
Using CATALINA_BASE:   /usr/tomcat8/tomcat2
Using CATALINA_HOME:   /usr/tomcat8
Using CATALINA_TMPDIR: /usr/tomcat8/tomcat2/temp
Using JRE_HOME:        /usr/lib/jvm/jre-1.8.0-openjdk-1.8.0.232.b09-0.el7_7.x86_64
Using CLASSPATH:       /usr/tomcat8/bin/bootstrap.jar:/usr/tomcat8/bin/tomcat-juli.jar
[root@centos7 bin]# ./stopTomcat1
Using CATALINA_BASE:   /usr/tomcat8/tomcat1
```

```
Using CATALINA_HOME:   /usr/tomcat8
Using CATALINA_TMPDIR: /usr/tomcat8/tomcat1/temp
Using JRE_HOME:        /usr/lib/jvm/jre-1.8.0-openjdk-1.8.0.232.b09-0.el7_7.x86_64
Using CLASSPATH:       /usr/tomcat8/bin/bootstrap.jar:/usr/tomcat8/bin/tomcat-juli.jar
[root@centos7 bin]# ps aux | grep tomcat
root     27318  0.0  0.0 112644   964 pts/0    R+   02:52   0:00 grep --color=auto tomcat
```

Le Cluster de Répartition de Charge avec Apache et mod_jk

Modifiez le fichier **/etc/httpd/conf/workers.properties** :

```
[root@centos7 bin]# vi /etc/httpd/conf/workers.properties
[root@centos7 bin]# cat /etc/httpd/conf/workers.properties
worker.list=balancer

worker.tomcat1.type=ajp13
worker.tomcat1.host=10.0.2.51
worker.tomcat1.port=8109
worker.tomcat1.lbfactor=1

worker.tomcat2.type=ajp13
worker.tomcat2.host=10.0.2.51
worker.tomcat2.port=8209
worker.tomcat2.lbfactor=1

worker.balancer.type=lb
worker.balancer.balance_workers=tomcat1,tomcat2
worker.balancer.sticky_session=1
```

Modifiez la section concernant Tomcat dans le fichier **/etc/httpd/conf/httpd.conf** et commentez la ligne **IncludeOptional conf.d/*.conf** :

```
[root@centos7 bin]# vi /etc/httpd/conf/httpd.conf
[root@centos7 bin]# tail /etc/httpd/conf/httpd.conf
```

```
#
# Load config files in the "/etc/httpd/conf.d" directory, if any.
# IncludeOptional conf.d/*.conf

LoadModule  jk_module     modules/mod_jk.so
JkWorkersFile   conf/workers.properties
JkLogFile   logs/mod_jk.log
JkLogLevel  info
JkMount     /docs/*    balancer
JkMount     /docs    balancer
```

Modifiez la section <Engine> du fichier **$CATALINA_HOME/tomcat1/conf/server.xml** :

```
[root@centos7 bin]# vi $CATALINA_HOME/tomcat1/conf/server.xml
...
    <Engine name="Catalina" defaultHost="localhost" jvmRoute="tomcat1">
...
```

Modifiez ensuite la section <Engine> du fichier **$CATALINA_HOME/tomcat2/conf/server.xml** :

```
[root@centos7 bin]# vi $CATALINA_HOME/tomcat2/conf/server.xml
...
    <Engine name="Catalina" defaultHost="localhost" jvmRoute="tomcat2">
...
```

Pour pouvoir tester la configuration, remplacer les fichiers index.html de chaque application **docs** afin de pouvoir identifier quelle instance répond à des requêtes :

```
[root@centos7 bin]# mv $CATALINA_HOME/tomcat1/webapps/docs/index.html
$CATALINA_HOME/tomcat1/webapps/docs/index.old
[root@centos7 bin]# vi $CATALINA_HOME/tomcat1/webapps/docs/index.html
[root@centos7 bin]# cat $CATALINA_HOME/tomcat1/webapps/docs/index.html
<html>
<title>Tomcat1</title>
```

```
<body>
<center>This is Tomcat1</center>
</body>
</html>
[root@centos7 bin]# mv $CATALINA_HOME/tomcat2/webapps/docs/index.html
$CATALINA_HOME/tomcat2/webapps/docs/index.old
[root@centos7 bin]# vi $CATALINA_HOME/tomcat2/webapps/docs/index.html
[root@centos7 bin]# cat $CATALINA_HOME/tomcat2/webapps/docs/index.html
<html>
<title>Tomcat2</title>
<body>
<center>This is Tomcat2</center>
</body>
</html>
[root@centos7 bin]#
```

Redémarrez le service httpd.service :

```
[root@centos7 bin]# systemctl restart httpd.service
```

Démarrez les deux instances de Tomcat :

```
[root@centos7 bin]# ./startTomcat1
Using CATALINA_BASE:   /usr/tomcat8/tomcat1
Using CATALINA_HOME:   /usr/tomcat8
Using CATALINA_TMPDIR: /usr/tomcat8/tomcat1/temp
Using JRE_HOME:        /usr/lib/jvm/jre-1.8.0-openjdk-1.8.0.232.b09-0.el7_7.x86_64
Using CLASSPATH:       /usr/tomcat8/bin/bootstrap.jar:/usr/tomcat8/bin/tomcat-juli.jar
Tomcat started.
[root@centos7 bin]# ./startTomcat2
Using CATALINA_BASE:   /usr/tomcat8/tomcat2
Using CATALINA_HOME:   /usr/tomcat8
Using CATALINA_TMPDIR: /usr/tomcat8/tomcat2/temp
Using JRE_HOME:        /usr/lib/jvm/jre-1.8.0-openjdk-1.8.0.232.b09-0.el7_7.x86_64
```

```
Using CLASSPATH:       /usr/tomcat8/bin/bootstrap.jar:/usr/tomcat8/bin/tomcat-juli.jar
Tomcat started.
```

Utilisez Lynx pour vous connecter à l'application **docs** :

```
[root@centos7 httpd]# lynx --dump http://www.i2tch.loc/docs
                              This is Tomcat2


[root@centos7 httpd]# lynx --dump http://www.i2tch.loc/docs
                              This is Tomcat2


[root@centos7 httpd]# lynx --dump http://www.i2tch.loc/docs
                              This is Tomcat2
```

Attention : Notez que l'affinité de session est activée par défaut par le module AJP.

Arrêtez maintenant l'instance **tomcat2** :

```
[root@centos7 bin]# ./stopTomcat2
Using CATALINA_BASE:   /usr/tomcat8/tomcat2
Using CATALINA_HOME:   /usr/tomcat8
Using CATALINA_TMPDIR: /usr/tomcat8/tomcat2/temp
Using JRE_HOME:        /usr/lib/jvm/jre-1.8.0-openjdk-1.8.0.232.b09-0.el7_7.x86_64
Using CLASSPATH:       /usr/tomcat8/bin/bootstrap.jar:/usr/tomcat8/bin/tomcat-juli.jar
```

Connectez-vous de nouveau à l'application **docs** :

```
[root@centos7 bin]# lynx --dump http://www.i2tch.loc/docs
                              This is Tomcat1
```

```
[root@centos7 bin]#
```

Important - Notez que c'est maintenant l'instance tomcat1 qui répond.

Le Cluster de Répartition de Charge avec Apache et mod_proxy_ajp

Vérifiez que les lignes **LoadModule proxy_ajp_module modules/mod_proxy_ajp.so**, **LoadModule proxy_balancer_module modules/mod_proxy_balancer.so** et **LoadModule proxy_module modules/mod_proxy.so** soient présentes dans le fichier **/etc/httpd/conf.modules.d/00-proxy.conf** :

```
[root@centos7 bin]# cat /etc/httpd/conf.modules.d/00-proxy.conf
# This file configures all the proxy modules:
LoadModule proxy_module modules/mod_proxy.so
LoadModule lbmethod_bybusyness_module modules/mod_lbmethod_bybusyness.so
LoadModule lbmethod_byrequests_module modules/mod_lbmethod_byrequests.so
LoadModule lbmethod_bytraffic_module modules/mod_lbmethod_bytraffic.so
LoadModule lbmethod_heartbeat_module modules/mod_lbmethod_heartbeat.so
LoadModule proxy_ajp_module modules/mod_proxy_ajp.so
LoadModule proxy_balancer_module modules/mod_proxy_balancer.so
LoadModule proxy_connect_module modules/mod_proxy_connect.so
LoadModule proxy_express_module modules/mod_proxy_express.so
LoadModule proxy_fcgi_module modules/mod_proxy_fcgi.so
LoadModule proxy_fdpass_module modules/mod_proxy_fdpass.so
LoadModule proxy_ftp_module modules/mod_proxy_ftp.so
LoadModule proxy_http_module modules/mod_proxy_http.so
LoadModule proxy_scgi_module modules/mod_proxy_scgi.so
LoadModule proxy_wstunnel_module modules/mod_proxy_wstunnel.so
```

Modifiez le fichier **/etc/httpd/conf/httpd.conf** :

```
[root@centos7 bin]# tail -n 15 /etc/httpd/conf/httpd.conf

#LoadModule jk_module     modules/mod_jk.so
#JkWorkersFile  conf/workers.properties
#JkLogFile  logs/mod_jk.log
#JkLogLevel info
#JkMount        /docs/*    balancer
#JkMount        /docs    balancer

ProxyTimeout 300

<Proxy balancer://tomcat8-docs>
    BalancerMember ajp://localhost:8109/docs route=tomcat1
    BalancerMember ajp://localhost:8209/docs route=tomcat2
</Proxy>

ProxyPass       /docs    balancer://tomcat8-docs
ProxyPassReverse    /docs    balancer://tomcat8-docs
```

Redémarrez le serveur httpd :

```
[root@centos7 bin]# systemctl restart httpd.service
```

Démarrez l'instance tomcat2 de Tomcat :

```
[root@centos7 bin]# ./startTomcat2
Using CATALINA_BASE:   /usr/tomcat8/tomcat2
Using CATALINA_HOME:   /usr/tomcat8
Using CATALINA_TMPDIR: /usr/tomcat8/tomcat2/temp
Using JRE_HOME:        /usr/lib/jvm/jre-1.8.0-openjdk-1.8.0.232.b09-0.el7_7.x86_64
Using CLASSPATH:       /usr/tomcat8/bin/bootstrap.jar:/usr/tomcat8/bin/tomcat-juli.jar
Tomcat started
```

Utilisez Lynx pour vous connecter à l'application **docs** :

```
[root@centos7 bin]# lynx --dump http://www.i2tch.loc/docs
                              This is Tomcat1


[root@centos7 bin]# lynx --dump http://www.i2tch.loc/docs
                              This is Tomcat2


[root@centos7 bin]# lynx --dump http://www.i2tch.loc/docs
                              This is Tomcat1


[root@centos7 bin]# lynx --dump http://www.i2tch.loc/docs
                              This is Tomcat2
```

Attention : Notez que l'affinité de session n'est pas activée par défaut par le module proxy.

Afin de mettre en place l'affinité de session, il convient d'utiliser un cookie appelé **ROUTEID**.

Modifiez le fichier **/etc/httdp/conf/httpd.conf** ainsi :

```
...
#IncludeOptional conf.d/*.conf

#LoadModule     jk_module       modules/mod_jk.so
#JkWorkersFile  conf/workers.properties
#JkLogFile      logs/mod_jk.log
#JkLogLevel     info
```

```
#JkMount                /docs/* balancer
#JkMount                /docs   balancer
Header add Set-Cookie "ROUTEID=.%{BALANCER_WORKER_ROUTE}e; path=/" env=BALANCER_ROUTE_CHANGED
<Proxy balancer://tomcat8-docs>
        BalancerMember ajp://localhost:8109/docs route=tomcat1
        BalancerMember ajp://localhost:8209/docs route=tomcat2
        ProxySet stickysession=ROUTEID
</Proxy>

ProxyPass               /docs   balancer://tomcat8-docs
ProxyPassReverse        /docs   balancer://tomcat8-docs
```

Testez ensuite l'affinité de session en utilisant un navigateur graphique.

Pour plus d'information concernant l'utilisation de mod_proxy, consultez **cette page**

Le Cluster en mode Maître/Esclave

La configuration en mode **Maître/Esclave** utilise le module **mod_jk**. Editez donc votre fichier **/etc/httpd/conf/httpd.conf** :

```
[root@centos7 bin]# vi /etc/httpd/conf/httpd.conf
[root@centos7 bin]# tail -n 20 /etc/httpd/conf/httpd.conf
EnableSendfile on

# Supplemental configuration
#
# Load config files in the "/etc/httpd/conf.d" directory, if any.
# IncludeOptional conf.d/*.conf

LoadModule  jk_module     modules/mod_jk.so
JkWorkersFile   conf/workers.properties
JkLogFile   logs/mod_jk.log
JkLogLevel  info
```

```
JkMount     /docs/*    balancer
JkMount     /docs    balancer

#<Proxy balancer://tomcat8-docs>
#   BalancerMember ajp://localhost:8109/docs route=tomcat1
#   BalancerMember ajp://localhost:8209/docs route=tomcat2
#</Proxy>

#ProxyPass      /docs    balancer://tomcat8-docs
#ProxyPassReverse   /docs    balancer://tomcat8-docs
```

Éditez ensuite le fichier **/etc/httpd/conf/workers.properties** :

```
[root@centos7 bin]# vi /etc/httpd/conf/workers.properties
[root@centos7 bin]# cat /etc/httpd/conf/workers.properties
worker.list=tomcat1,tomcat2,balancer

worker.tomcat1.type=ajp13
worker.tomcat1.host=10.0.2.51
worker.tomcat1.port=8109
# Indique que tomcat2 doit prendre le relais en cas de défaillance de tomcat1
worker.tomcat1.redirect=tomcat2

worker.tomcat2.type=ajp13
worker.tomcat2.host=10.0.2.51
worker.tomcat2.port=8209
# Indique que l'instance tomcat2 est un escalve
worker.tomcat2.activation=disabled

worker.balancer.type=lb
worker.balancer.balance_workers=tomcat1,tomcat2
```

Redémarrez le serveur httpd :

```
[root@centos7 bin]# systemctl restart httpd
```

Utilisez Lynx pour vous connecter à l'application **docs** :

```
[root@centos7 bin]# lynx --dump http://www.i2tch.loc/docs
                              This is Tomcat1


[root@centos7 bin]# lynx --dump http://www.i2tch.loc/docs
                              This is Tomcat1


[root@centos7 bin]# lynx --dump http://www.i2tch.loc/docs
                              This is Tomcat1


[root@centos7 bin]#
```

Arrêtez l'instance tomcat1 :

```
[root@centos7 bin]# ./stopTomcat1
Using CATALINA_BASE:   /usr/tomcat8/tomcat1
Using CATALINA_HOME:   /usr/tomcat8
Using CATALINA_TMPDIR: /usr/tomcat8/tomcat1/temp
Using JRE_HOME:        /usr/lib/jvm/jre-1.8.0-openjdk-1.8.0.232.b09-0.el7_7.x86_64
Using CLASSPATH:       /usr/tomcat8/bin/bootstrap.jar:/usr/tomcat8/bin/tomcat-juli.jar
```

Utilisez de nouveau Lynx pour vous connecter à l'application **docs** :

```
[root@centos7 bin]# lynx --dump http://www.i2tch.loc/docs
                              This is Tomcat2
```

```
[root@centos7 bin]# lynx --dump http://www.i2tch.loc/docs
                              This is Tomcat2


[root@centos7 bin]#
```

Attention : Notez que le basculement est automatique en cas de défaillance de l'instance tomcat1.

Maintenir l'Etat des Clients

Préparation

Editez le fichier **web.xml** de l'application **/docs** de chaque instance de Tomcat en incluant la directive **<distributable/>** :

```
[root@centos7 bin]# vi $CATALINA_HOME/tomcat1/webapps/docs/WEB-INF/web.xml
[root@centos7 bin]# vi $CATALINA_HOME/tomcat2/webapps/docs/WEB-INF/web.xml
[root@centos7 bin]# tail $CATALINA_HOME/tomcat2/webapps/docs/WEB-INF/web.xml
                      http://xmlns.jcp.org/xml/ns/javaee/web-app_3_1.xsd"
  version="3.1"
  metadata-complete="true">

  <display-name>Tomcat Documentation</display-name>
  <description>
     Tomcat Documentation.
  </description>
 <distributable/>
</web-app>
[root@centos7 bin]# tail $CATALINA_HOME/tomcat1/webapps/docs/WEB-INF/web.xml
                      http://xmlns.jcp.org/xml/ns/javaee/web-app_3_1.xsd"
```

```
  version="3.1"
  metadata-complete="true">

  <display-name>Tomcat Documentation</display-name>
  <description>
     Tomcat Documentation.
  </description>
 <distributable/>
</web-app>
```

Créez les fichiers **$CATALINA_HOME/tomcat1/webapps/docs/session.jsp** et **$CATALINA_HOME/tomcat2/webapps/docs/session.jsp** :

```
[root@centos7 bin]# vi $CATALINA_HOME/tomcat1/webapps/docs/session.jsp
[root@centos7 bin]# cat $CATALINA_HOME/tomcat1/webapps/docs/session.jsp
<%@page language="java" %>
<html>
<body>
<h3>
Session : <%= session.getId() %>
</h3>
</body>
</html>
```

```
[root@centos7 bin]# vi $CATALINA_HOME/tomcat2/webapps/docs/session.jsp
[root@centos7 bin]# cat $CATALINA_HOME/tomcat2/webapps/docs/session.jsp
<%@page language="java" %>
<html>
<body>
<h3>
Session : <%= session.getId() %>
</h3>
</body>
</html>
```

Décommentez la ligne suivante dans les fichiers **server.xml** :

```
[root@centos7 bin]# vi $CATALINA_HOME/tomcat1/conf/server.xml
...
        <Cluster className="org.apache.catalina.ha.tcp.SimpleTcpCluster"/>
...
[root@centos7 bin]# vi $CATALINA_HOME/tomcat2/conf/server.xml
...
        <Cluster className="org.apache.catalina.ha.tcp.SimpleTcpCluster"/>
...
```

Sessions Persistentes sur Système de Fichiers

Editez maintenant les fichier **$CATALINA_HOME/tomcat1/conf/context.xml** et **$CATALINA_HOME/tomcat2/conf/context.xml** en ajoutant la section suivante :

```
    <Manager className="org.apache.catalina.session.PersistentManager" >
        <Store className="org.apache.catalina.session.FileStore"
        directory="/tmp/sessions/" />
    </Manager>
```

Vous obtiendrez un résultat similaire à celui-ci :

```
[root@centos7 bin]# cat /usr/tomcat8/tomcat1/conf/context.xml
<?xml version='1.0' encoding='utf-8'?>
<!--
  Licensed to the Apache Software Foundation (ASF) under one or more
  contributor license agreements.  See the NOTICE file distributed with
  this work for additional information regarding copyright ownership.
  The ASF licenses this file to You under the Apache License, Version 2.0
  (the "License"); you may not use this file except in compliance with
  the License.  You may obtain a copy of the License at
```

```
      http://www.apache.org/licenses/LICENSE-2.0

  Unless required by applicable law or agreed to in writing, software
  distributed under the License is distributed on an "AS IS" BASIS,
  WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
  See the License for the specific language governing permissions and
  limitations under the License.
-->
<!-- The contents of this file will be loaded for each web application -->
<Context>

    <Manager className="org.apache.catalina.session.PersistentManager" >
        <Store className="org.apache.catalina.session.FileStore"
        directory="/tmp/sessions/" />
    </Manager>

    <!-- Default set of monitored resources. If one of these changes, the    -->
    <!-- web application will be reloaded.                                   -->
    <WatchedResource>WEB-INF/web.xml</WatchedResource>
    <WatchedResource>${catalina.base}/conf/web.xml</WatchedResource>

    <!-- Uncomment this to disable session persistence across Tomcat restarts -->
    <!--
    <Manager pathname="" />
    -->

    <!-- Uncomment this to enable Comet connection tacking (provides events
         on session expiration as well as webapp lifecycle) -->
    <!--
    <Valve className="org.apache.catalina.valves.CometConnectionManagerValve" />
    -->
</Context>
```

Créez le répertoire **/tmp/sessions** pour contenir les fichiers de sessions :

```
[root@centos7 bin]# mkdir /tmp/sessions
```

En utilisant votre navigateur graphique, saisissez l'URL suivante :

```
http://www.i2tch.loc/docs/session.jsp
```

Vous obtiendrez une résultat similaire à l'exemple suivant :

```
Session : 7DA9FEE977543F1F574DADFA7B1FADD0.tomcat1
```

ou

```
Session : 7DA9FEE977543F1F574DADFA7B1FADD0.tomcat2
```

Selon l'instance de Tomcat qui a répondu, arrêtez cette instance :

```
[root@centos7 bin]# ./stopTomcat1
Using CATALINA_BASE:   /usr/tomcat8/tomcat1
Using CATALINA_HOME:   /usr/tomcat8
Using CATALINA_TMPDIR: /usr/tomcat8/tomcat1/temp
Using JRE_HOME:        /usr/lib/jvm/jre-1.8.0-openjdk-1.8.0.232.b09-0.el7_7.x86_64
Using CLASSPATH:       /usr/tomcat8/bin/bootstrap.jar:/usr/tomcat8/bin/tomcat-juli.jar
```

ou

```
[root@centos7 bin]# ./stopTomcat2
Using CATALINA_BASE:   /usr/tomcat8/tomcat2
Using CATALINA_HOME:   /usr/tomcat8
Using CATALINA_TMPDIR: /usr/tomcat8/tomcat2/temp
Using JRE_HOME:        /usr/lib/jvm/jre-1.8.0-openjdk-1.8.0.91-1.b14.el7_2.x86_64
Using CLASSPATH:       /usr/tomcat8/bin/bootstrap.jar:/usr/tomcat8/bin/tomcat-juli.jar
```

Contrôlez le contenu du répertoire **/tmp/sessions** :

```
[root@centos7 bin]# ls -l /tmp/sessions
total 4
-rw-r--r-- 1 root root 263 Jul  5 23:32 7DA9FEE977543F1F574DADFA7B1FADD0.tomcat1.session
```

Revenez à votre navigateur Web graphique et rafraîchissez la page. Vous obtiendrez un résultat démontrant que la session est resté la même malgré le fait que c'est l'autre instance de Tomcat qui vous a répondu.

```
Session : 7DA9FEE977543F1F574DADFA7B1FADD0.tomcat1
```

ou

```
Session : 7DA9FEE977543F1F574DADFA7B1FADD0.tomcat2
```

SER307 - Validation de la Formation

Contenu du Module

- **SER307 - Validation de la Formation**
  - Contenu du Module
  - Pour Aller Plus Loin
    - Support de Cours
  - Rappel du Programme de la Formation
    - Jour #1
    - Jour #2
    - Jour #3
  - Évaluation de la Formation
  - Validation des Acquis

Pour Aller Plus Loin

Support de Cours

L'accès au supports de cours ainsi que les LABS et les validations des acquis se fait grâce à un abonnement annuel par stagiaire à une plateforme de cours sur Internet.

L'utilisation de cette plateforme permet :

- de mesurer le niveau du stagiaire avant la formation et celui atteint en fin de formation grâce aux tests de validations des acquis,
- de suivre du travail de chaque participant en termes de temps passé dans chaque module grâce à un reporting détaillé.

L'abonnement permet aux stagiaires :

- de télécharger des supports de cours et des LABS au format PDF le dernier jour de la formation,
- de refaire les LABS en mode autonome en cas de missions décalées en relation avec le contenu de la formation initiale,
- de rester en contact avec le formateur en cas de problèmes en production liés au contenu du cours,
- de consulter les mises à jour du contenu des supports de cours pendant la période de l'abonnement,
- d'échanger avec les autres participants de la session ainsi qu'avec les anciens stagiaires.

Rappel du Programme de la Formation

Jour #1

- **SER300 - Administration d'un serveur d'applications JEE avec Tomcat**
  - Contenu du Module
  - Prérequis
    - Matériel
    - Logiciels
    - Internet
  - Utilisation de l'Infrastructure

    - Programme de la Formation

- **SER301 - Présentation des Technologies** - 2 heures.
    - Présentation de Tomcat 8
        - Historique et différentes versions
    - Rappel sur les applications Web en Java
    - Contenu statique, dynamique, Servlets, JSPs et Composants EJB
        - Servlets
        - JSP
        - Enterprise JavaBeans - EJB
    - Le Modèle MVC
    - Les Modules Java EE
        - Modules Web
        - Modules EJB
        - Modules Clients
        - Modules de Connecteurs
    - Positionnement d'Apache Tomcat dans la norme Java EE
        - Structure d'une Application Web
        - Le Descripteur de Déploiement web.xml
        - Les Sessions HTTP

- **SER302 - Installation de Tomcat 8 et les serveurs associés** - 2 heures.
    - Désactiver SELinux
    - Tomcat et JDK
    - Apache
        - Présentation d'Apache
        - Installation
        - Testez le serveur apache avec telnet
    - Coupler Tomcat et Apache
    - MariaDB
        - Présentation
        - Installation
        - Configuration
    - OpenLDAP
        - Présentation

        - Installation

- **SER303 - Configuration du serveur Tomcat 8** - 2 heures.
    - Architecture du Serveur
    - Fichiers de Configuration
        - Le Fichier $CATALINA_HOME/conf/server.xml
            - L'élément <Server>
            - L'élément <Service>
            - L'élément <Connector>
            - L'élément <Executor>
            - L'élément <Engine>
            - L'élément <Host>
            - L'élément <Context>
            - L'élément <Realm>
            - L'élément <Loader>
            - L'élément <Manager>
            - L'élément <Store>
            - L'élément <Valve>
                - Filtrage de l'adresse IP
                - Filtrage de nom de la machine du client
                - LAB #1 -Journalisation des Requêtes Client dans un Fichier Texte
                - LAB #2 -Journalisation des Requêtes Client dans une Base de Données
            - L'élément <Listener>
        - Le Fichier $CATALINA_HOME/conf/web.xml
        - Le Fichier $CATALINA_HOME/conf/tomcat-users.xml
        - Le Fichier $CATALINA_HOME/conf/catalina.policy
        - Configuration des Ressources
            - Portée des Ressources
            - Pools de Connexion
            - Sessions JavaMail
            - JavaBeans
            - Entrées D'Environnement

Jour #2

- **SER304 - Déploiement et Gestion des Applications** - 3 heures.
  - Déployer une application
  - Déploiement Automatique
  - L'Élément Context
  - Déploiement avec XML
  - Application Manager de Tomcat
    - L'interface Texte
      - list
      - deploy
      - start
      - stop
      - reload
      - undeploy
      - resources
      - serverinfo
    - L'interface HTML
    - L'interface ANT
  - Deployer de Tomcat
- **SER305 - Sécurité du serveur Tomcat 8** - 4 heures.
  - Authentification, Autorisation et Cryptage
    - Authentification
    - Autorisation
    - Cryptage
  - La Sécurité sous Tomcat
  - Configuration
    - Realms
      - User Database Realm
      - JDBC Realm
      - DataSource Realm
      - JNDI Realm
        - Le format LDIF

            - La commande ldapadd
        - JAAS Realm
        - Combined Realm
        - LockOut Realm
    - Tomcat et le SSO
    - Tomcat et le SSL
        - Présentation de SSL
            - Fonctionnement de SSL
        - Configurer Tomcat
        - Configurer Apache
            - Installation de SSL
            - Configuration de SSL
            - Mise en place des paramètres de sécurité SSL
            - Tester Votre Configuration
        - Apache en Frontal HTTPS
        - Restrictions d'Accès
        - Le Gestionnaire de Sécurité

Jour #3

- **SER306 - Journalisation, Supervision et Clustering** - 6 heures.
    - Configuration des journaux
        - java.util.logging
        - log4j
    - Supervision
        - JMeter
        - Interface JMX
        - JConsole
    - Clustering avec Tomcat
        - Préparation
        - Le Cluster de Répartition de Charge avec Apache et mod_jk
        - Le Cluster de Répartition de Charge avec Apache et mod_proxy_ajp
        - Le Cluster en mode Maître/Esclave

        - Maintenir l'Etat des Clients
            - Préparation
            - Sessions Persistantes sur Système de Fichiers

- **SER307 - Validation de la Formation** - 1 heure.
    - Pour Aller Plus Loin
        - Support de Cours
        - L'Infrastructure Hors Formation
            - Matériel
            - Logiciels
            - Machine Virtuelle
    - Rappel du Programme de la Formation
        - Jour #1
        - Jour #2
        - Jour #3
    - Remettre en Etat l'Infrastructure
    - Évaluation de la Formation
    - Évaluation des Acquis
    - Remerciements

Évaluation de la Formation

Afin de valider votre formation, veuillez compléter l'Évaluation de la Formation et passer la Validation des Acquis.
