

Mise en œuvre d'un serveur Tomcat sous Linux

Support de cours





Module 0

A propos de cette formation

Votre formateur

- Etienne LANGLET
- Equipe développement & méthodologie
- Développement
 - Java, PHP, Python
- Administration de serveurs
 - Apache, Tomcat, WildFly, ...

Votre formation – Ressources à votre disposition

- Cette présentation
- Ressources de travaux pratiques
 - <https://github.com/elanglet/tomcat-linux>



Module 1

Java, les serveurs d'application, Tomcat

Java : historique et philosophie

- Langage simple, orienté objet, distribué, interprété, robuste, sécurisé, indépendant des architectures, portable, performant, multithread et dynamique
- Plate-forme (JDK 1.0) disponible depuis 1995
- Langage adopté par de nombreux éditeurs informatiques
 - Oracle, IBM, Google...
- Site dédié www.oracle.com/technetwork/java/
 - Nombreuses ressources disponibles : SDK, documentation, tutoriaux, exemples de codes
 - Communauté de développeurs très importante

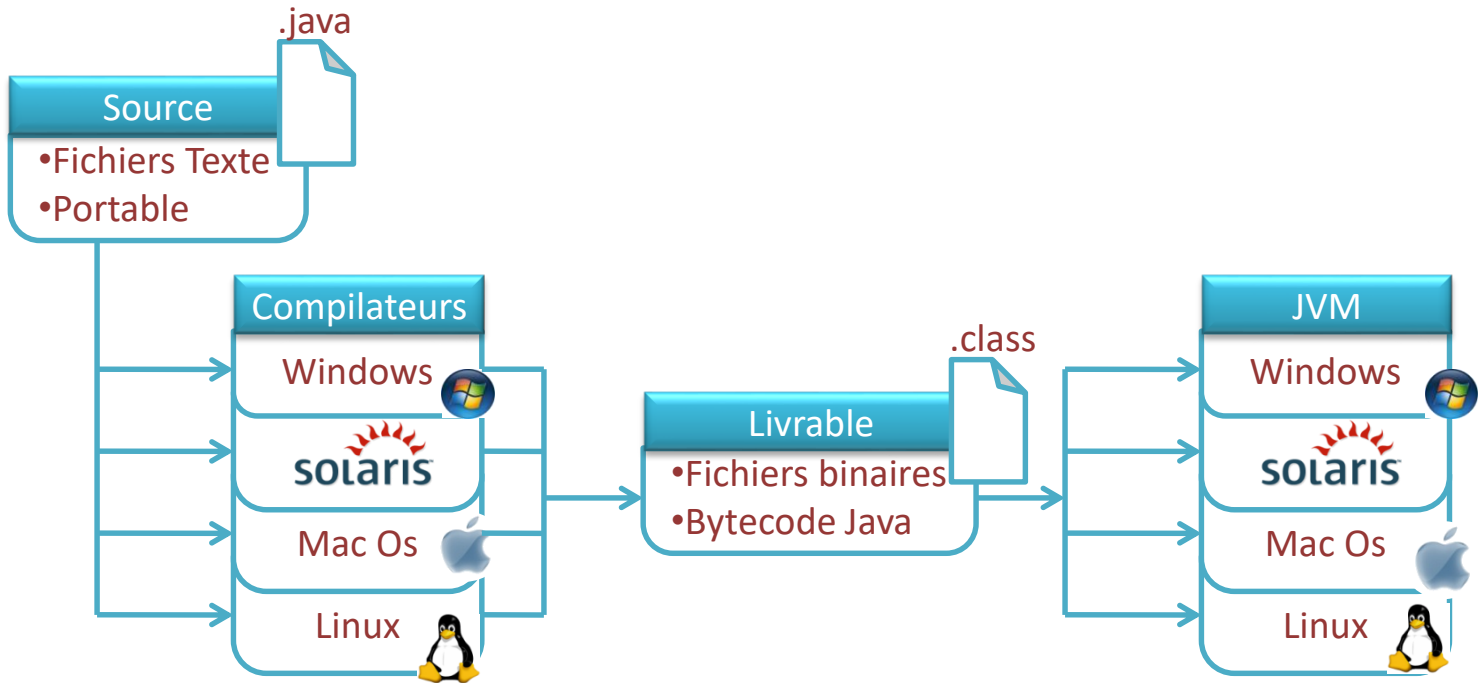


Les différentes plateformes Java

- Java SE (Java Standard Edition)
 - Appelée aussi JDK (Java Development Kit) ou SDK (Software Development Kit)
 - Développement d'applications clientes et serveurs et d'Applets
 - Contient l'ensemble des API de base (classes regroupées par packages) et des outils en ligne de commande pour développer
 - Fournit la Machine Virtuelle Java (JVM)
- Java EE (Java Enterprise Edition)
 - L'extension serveur de Java SE
 - Applications Internet/Intranet

Notion de JVM (Java Virtual Machine)

- Permet l'exécution d'une application Java sur n'importe quel système d'exploitation.



Qu'est ce que Java EE ?

- Extension de la plateforme standard Java (Java SE)
- Standard pour le développement d'applications Web en Java
- Périmètre :
 - Des API de programmation pour développer des applications et les packager en livrables.
 - Des serveurs d'applications capables d'exécuter ces applications.
- Les serveurs d'applications
 - Mettent en œuvre les applications Web développées avec Java EE
 - Utilisent HTTP pour la communication avec les clients : navigateurs
 - Références :
 - Tomcat (Apache Software Foundation)
 - WildFly (RedHat Software)
 - WebLogic (Oracle)
 - ...

Tomcat

- Tomcat est un serveur d'applications Java EE.
 - Apache Software Foundation -> « Apache Tomcat »
- Versions :
 - 2001 : Tomcat 3
 - 2020 : Tomcat 9
- Licence :
 - Licence Apache (Licence Libre & Open-Source)
- Tomcat fournit un environnement d'exécution pour les applications Web Java EE et il nécessite une JVM pour fonctionner (il est écrit en Java !)



Travaux pratiques

- Exercice <n°> : Titre de l'exercice>
 - <Enoncé>

Fin du module

- Résumé de ce qui a été vu dans ce module
 - <Sujet 1>
 - <Sujet N>
- Rapide test de connaissances
 - <Question 1>
 - <Question 5>
- Avez-vous des questions ?



Module 2

Installation de Tomcat

Prérequis

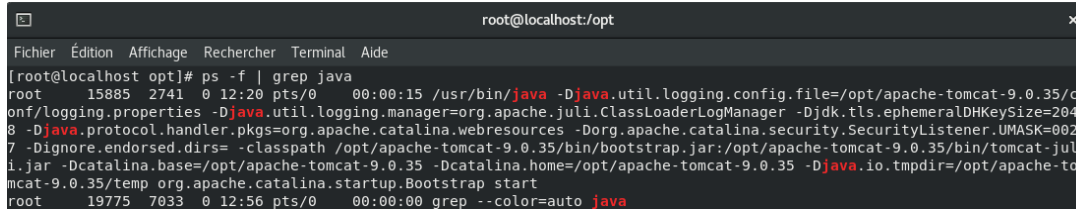
- Tomcat est écrit en Java
 - Il nécessite une JVM pour fonctionner
- L'installation d'un JDK est donc nécessaire !
 - OpenJDK 11 fournit une implémentation libre de droit de la plateforme Java

Installation de Tomcat

- Une simple archive à décompresser !
- Téléchargement :
 - <http://tomcat.apache.org>
- Archives :
 - Format ZIP
 - Format .TAR.GZ
- Installation :
 - Décompression de l'archive.
 - Création de la variable d'environnement CATALINA_HOME pointant sur le répertoire d'installation de Tomcat.

Démarrage et arrêt de Tomcat

- \$CATALINA_HOME qualifie le répertoire de base de l'installation de Tomcat
- Démarrage
 - \$CATALINA_HOME/bin/startup.(sh|bat)
- Arrêt
 - \$CATALINA_HOME/bin/shutdown.(sh|bat)
- Vérification du démarrage :
 - Commande ps dans un terminal :

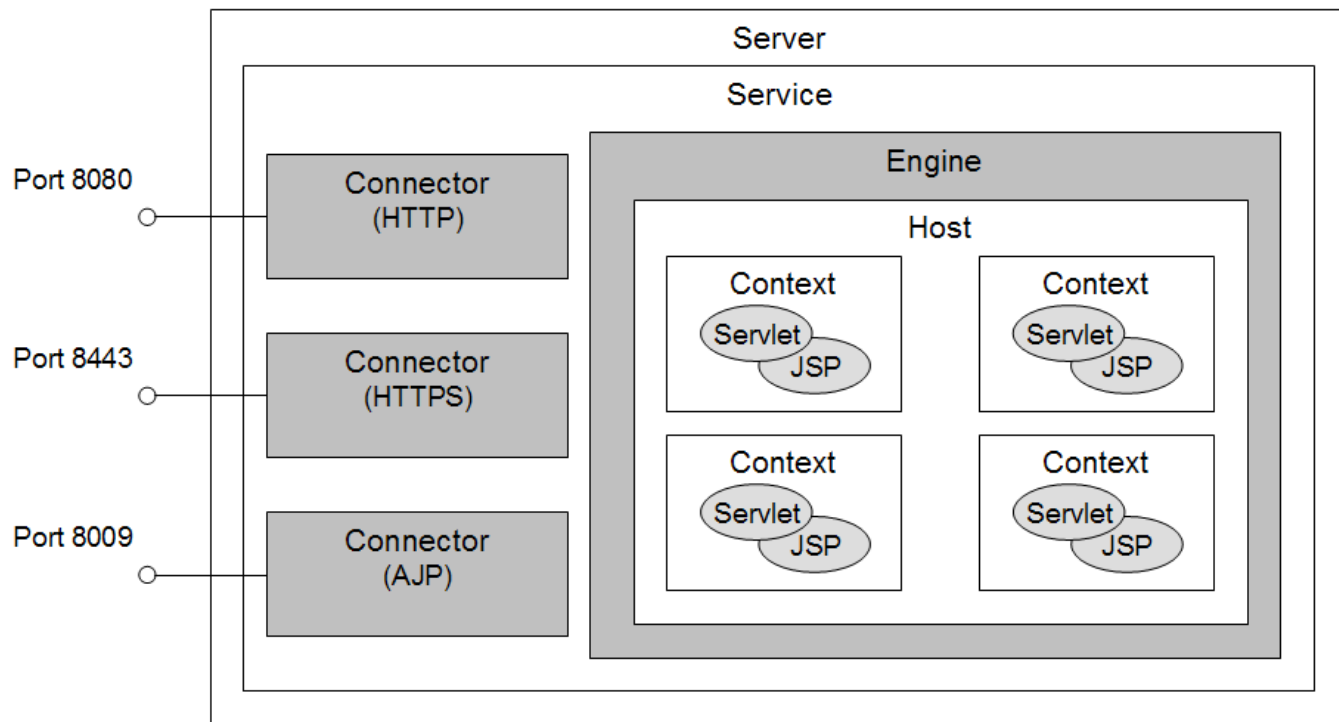


```
root@localhost:opt
Fichier  Édition  Affichage  Rechercher  Terminal  Aide
[root@localhost opt]# ps -f | grep java
root    15885   2741   0 12:20 pts/0    00:00:15 /usr/bin/java -Djava.util.logging.config.file=/opt/apache-tomcat-9.0.35/conf/logging.properties -Djava.util.logging.manager=org.apache.juli.ClassLoaderLogManager -Djdk.tls.ephemeralDHKeySize=2048 -Djava.protocol.handler.pkgs=org.apache.catalina.webresources -Dorg.apache.catalina.security.SecurityListener.UMASK=0027 -Dignore.endorsed.dirs=-classpath /opt/apache-tomcat-9.0.35/bin/bootstrap.jar:/opt/apache-tomcat-9.0.35/bin/tomcat-juli.jar -Dcatalina.base=/opt/apache-tomcat-9.0.35 -Dcatalina.home=/opt/apache-tomcat-9.0.35 -Djava.io.tmpdir=/opt/apache-tomcat-9.0.35/temp org.apache.catalina.startup.Bootstrap start
root    19775   7033   0 12:56 pts/0    00:00:00 grep --color=auto java
```

- <http://localhost:8080> dans un navigateur Web

Architecture de Tomcat

- Dans le vocabulaire Tomcat « Context » représente une application déployée.



Arborescence

- bin/
 - Scripts et fichiers de démarrage de Tomcat
- conf/
 - Fichiers de configuration, notamment le fichier principal : `server.xml`
- lib/
 - Bibliothèques Java accessibles par Tomcat et les applications déployées
- logs/
 - Fichiers journaux
- temp/
 - Fichiers temporaires utilisables par les applications déployées
- webapps/
 - Répertoire d'installation des applications
- work/
 - Répertoire de traitement interne à Tomcat

Un serveur réseau...

- Ports utilisés par Tomcat :
 - 8080 : Accès en HTTP par les clients.
 - 8009 : Accès en AJP (Apache JServ Protocol) depuis un frontal Web Apache (Reverse-Proxy).
 - 8005 : Arrêt du serveur.
- Les valeurs de ces ports sont reconfigurables dans le fichier `server.xml`.

TP : Installation de Tomcat

Travaux pratiques

- Installation de la plateforme Java
 - Installation de Java (CentOs)
 - `# dnf install java-11-openjdk-devel`
 - Vérification
 - `# java -version`
- Installation de Tomcat
 - Téléchargement de Tomcat
 - Fichier **apache-tomcat-x.x.x.tar.gz**
 - Décompression
 - `# tar xvzf apache-tomcat-x.x.x.tar.gz`
 - Déplacer le répertoire d'installation pour Tomcat
 - `# mv apache-tomcat-x.x.x /opt/`
 - Créer un compte utilisateur Tomcat
 - `# useradd -r tomcat`
 - Changer le propriétaire de l'arborescence
 - `# chown -R tomcat:tomcat /opt/apache-tomcat-x.x.x`
 - Définir et exporter la variable CATALINA_HOME dans le fichier **/etc/profile**
 - **`export CATALINA_HOME=/opt/apache-tomcat-x.x.x`**
 - Recharger l'environnement
 - `# source /etc/profile`

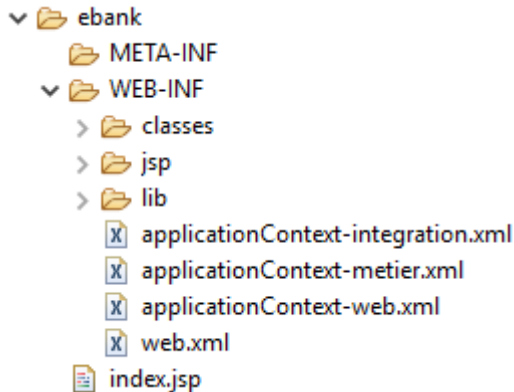


Module 3

Déploiement d'applications

Concepts

- Les applications Web Java EE sont packagées dans des archives.
 - Fichiers .war
 - WAR = Web ARchive
 - Un fichier ZIP avec une arborescence très stricte !



Déploiement d'applications dans Tomcat

- Le déploiement d'applications peut se faire à chaud !
 - i.e. : Pas de redémarrage nécessaire !
- Copie du fichier **.war** dans le répertoire webapps/ de Tomcat.
 - Le fichier **.war** est ensuite décompressé à la volée.
- Analyse des journaux pour s'assurer du bon déploiement de l'application.
 - \$CATALINA_HOME/logs/catalina.out
- Accès à l'application :
 - Pour une archive webapps/app1.war
 - L'accès à l'application se fera par l'URL <http://localhost:8080/app1>

TP : Déploiement d'applications

Travaux pratiques

- Déployer l'application « ebank »
 - Récupération du fichier **ebank.war** de l'application
 - <https://github.com/elanglet/tomcat-linux>
 - Copie du fichier **ebank.war** dans le répertoire webapps/ de Tomcat.
 - Vérification des traces dans le fichier \$CATALINA_HOME/logs/catalina.out

```
org.apache.catalina.startup.HostConfig.deployWAR Déploiement de l'archive [/opt/apache-tomcat-9.0.35/webapps/ebank.war] de l'application web
org.apache.jasper.servlet.JspScanner.scanJars AU MOINS un fichier JAR a été analysé pour trouver des TLDs mais il n'en contenait pas, le mode "debug" du journal peut être ac
de JAR scannés sans succès ; éviter d'analyser des JARs inutilement peut améliorer sensiblement le temps de démarrage et le temps de compilation des JSPs
org.springframework.web.context.ContextLoader.initWebApplicationContext Root WebApplicationContext: initialization started
org.springframework.web.context.support.XmlWebApplicationContext.prepareRefresh Refreshing Root WebApplicationContext: startup date [Fri May 15 13:51:34 CEST 2020]; root of
org.springframework.beans.factory.xml.XmlBeanDefinitionReader.loadBeanDefinitions Loading XML bean definitions from ServletContext resource [/WEB-INF/applicationContext-inte
org.springframework.beans.factory.xml.XmlBeanDefinitionReader.loadBeanDefinitions Loading XML bean definitions from ServletContext resource [/WEB-INF/applicationContext-met
org.springframework.jdbc.datasource.DriverManagerDataSource.setDriverClassName Loaded JDBC driver: org.h2.Driver
org.hibernate.Version.logVersion HHH000412: Hibernate Core (5.3.7.Final)
org.hibernate.cfg.Environment.<clinit> HHH000206: hibernate.properties not found
org.hibernate.annotations.common.reflection.java.JavaReflectionManager.<clinit> HCANN000001: Hibernate Commons Annotations (5.0.4.Final)
org.hibernate.dialect.Dialect.<clinit> HHH000400: Using dialect: org.hibernate.dialect.H2Dialect
org.springframework.jdbc.datasource.init.ScriptUtils.executeSqlScript Executing SQL script from class path resource [fr/formation/banque/scripts/schema-banque-drop.sql]
org.springframework.jdbc.datasource.init.ScriptUtils.executeSqlScript Executed SQL script from class path resource [fr/formation/banque/scripts/schema-banque-drop.sql] in 2
org.springframework.jdbc.datasource.init.ScriptUtils.executeSqlScript Executing SQL script from class path resource [fr/formation/banque/scripts/schema-banque.sql]
org.springframework.jdbc.datasource.init.ScriptUtils.executeSqlScript Executed SQL script from class path resource [fr/formation/banque/scripts/schema-banque.sql] in 32 ms.
org.springframework.orm.hibernate5.HibernateTransactionManager.afterPropertiesSet Using DataSource [org.springframework.jdbc.datasource.DriverManagerDataSource@77a35b2f] of
TransactionManager
org.springframework.web.context.ContextLoader.initWebApplicationContext Root WebApplicationContext: initialization completed in 2701 ms
org.springframework.web.servlet.DispatcherServlet.initServletBean FrameworkServlet 'banque': initialization started
org.springframework.web.context.support.XmlWebApplicationContext.prepareRefresh Refreshing WebApplicationContext for namespace 'banque-servlet': startup date [Fri May 15 13:
licationContext
org.springframework.beans.factory.xml.XmlBeanDefinitionReader.loadBeanDefinitions Loading XML bean definitions from ServletContext resource [/WEB-INF/applicationContext-web.
org.springframework.web.servlet.mvc.annotation.DefaultAnnotationHandlerMapping.registerHandler Mapped URL path [/identification.do] onto handler 'identificationController'
org.springframework.web.servlet.mvc.annotation.DefaultAnnotationHandlerMapping.registerHandler Mapped URL path [/accueil.do] onto handler 'indexController'
org.springframework.web.servlet.mvc.annotation.DefaultAnnotationHandlerMapping.registerHandler Mapped URL path [/virement.do] onto handler 'virementController'
org.springframework.web.servlet.DispatcherServlet.initServletBean FrameworkServlet 'banque': initialization completed in 539 ms
org.apache.catalina.startup.HostConfig.deployWAR Le déploiement de l'archive de l'application web [/opt/apache-tomcat-9.0.35/webapps/ebank.war] s'est terminé en [5 249] ms
```

- L'application doit être accessible à l'URL <http://localhost:8080/ebank>



Module 4

Gestion du serveur Tomcat

Concepts

- Le serveur Tomcat fonctionne sur une Java Virtual Machine.
- Elle est responsable de l'interaction avec le système d'exploitation.
 - Et de la gestion de la mémoire !
- Trop peu de mémoire utilisable par la JVM conduira à une défaillance fatale de Tomcat.
- Tomcat fournit un outil de base pour observer les consommations mémoire.
 - Et bien d'autres paramètres !
- D'autres outils sur le marché peuvent fournir des informations similaires en temps réel.

Le gestionnaire de Tomcat

- Tomcat vient avec une application simple de gestion.
 - Le « manager »
- Le « manager » permet :
 - Le déploiement et la gestion des applications.
 - L'accès aux informations de consommation de ressources du serveur.
- Pour activer le « manager », il faut créer un compte d'accès dans le fichier `$CATALINA_HOME/conf/tomcat-users.xml`

```
<role rolename="manager-gui"/>  
<user username="admin" password="secret" roles="manager-gui" />
```

- Après un redémarrage du serveur, le « manager » est disponible à l'adresse <http://localhost:8080/manager>

Le manager de Tomcat pour la supervision des ressources

- L'onglet « Etat du serveur » permet de voir les ressources consommées par Tomcat.



Etat du serveur

Gestionnaire			
Lister les applications	Aide HTML Gestionnaire	Aide Gestionnaire	Etat complet du serveur

Serveur							
Version de serveur	Version de la JVM	Fournisseur de la JVM	Nom d'OS	Version d'OS	Architecture d'OS	Nom d'hôte	Adresse IP
Apache Tomcat/9.0.35	11.0.7+10-LTS	Oracle Corporation	Linux	4.18.0-147.8.1.el8_1.x86_64	amd64	localhost.localdomain	127.0.0.1

JVM

Mémoire disponible : 225.32 MB Mémoire totale : 289.00 MB Mémoire maximale : 1954.00 MB

Pool mémoire	Type	Initial	Total	Maximum	Utilisé
G1 Eden Space	Heap memory	8.00 MB	160.00 MB	-0.00 MB	33.00 MB
G1 Old Gen	Heap memory	116.00 MB	107.00 MB	1954.00 MB	7.67 MB (0%)
G1 Survivor Space	Heap memory	0.00 MB	22.00 MB	-0.00 MB	22.00 MB
CodeHeap 'non-nmethods'	Non-heap memory	2.43 MB	2.43 MB	5.55 MB	1.21 MB (21%)
CodeHeap 'non-profiled nmethods'	Non-heap memory	2.43 MB	2.43 MB	117.22 MB	2.10 MB (1%)
CodeHeap 'profiled nmethods'	Non-heap memory	2.43 MB	9.81 MB	117.21 MB	9.26 MB (7%)
Compressed Class Space	Non-heap memory	0.00 MB	5.62 MB	1024.00 MB	5.30 MB (0%)
Metaspace	Non-heap memory	0.00 MB	47.70 MB	-0.00 MB	46.10 MB

"http-nio-8080"

La mémoire de la JVM

JVM

Mémoire disponible : 225.32 MB Mémoire totale : 289.00 MB Mémoire maximale : 1954.00 MB

- Mémoire totale :
 - Mémoire actuellement réservée auprès de l'OS.
- Mémoire disponible :
 - Sur la quantité actuellement réservée auprès de l'OS.
- Mémoire maximale :
 - Quantité maximale réservable auprès de l'OS.
- Si la mémoire totale est égale à la mémoire maximale et que la mémoire disponible est faible, alors une défaillance fatale est proche.

Modification des allocations mémoires

- Il est possible de modifier la quantité de mémoire maximale utilisable par la JVM de Tomcat.
 - Sous réserve que les ressources physiques le permettent !
- Des options de démarrage Java peuvent être ajoutée pour spécifier :
 - La quantité de mémoire allouée au démarrage de la JVM.
 - La quantité de mémoire maximale allouable par la JVM
- Ces options doivent être spécifiées dans le fichier `$CATALINA_HOME/bin/startup.sh` grâce à la variable **JAVA_OPTS**.
- Exemple :
 - Pour une mémoire initiale à 1Go et une mémoire maximale à 2Go

```
export JAVA_OPTS="-Xms1024m -Xmx2048m"
```

TP : Gestion du serveur

Travaux pratiques

- Activer le manager de Tomcat
 - Modifier le fichier `$CATALINA_HOME/conf/tomcat-users.xml`
 - Déclarer le rôle « manager-gui ».
 - Créer un compte possédant ce rôle.
 - Redémarrer Tomcat et vérifier que l'accès au manager est possible :
 - <http://localhost:8080/manager>
- Reconfigurer la JVM de Tomcat
 - Modifier les allocations mémoire pour :
 - Une valeur initiale à 512Mo
 - Une valeur maximale à 2Go
 - Redémarrer Tomcat et vérifier les modifications grâce au manager.



Fin de la formation

Mise en œuvre d'un serveur Tomcat sous Linux

Pour aller plus loin

- Les cours associés
 - T473-010 - Tomcat – Administrer un serveur d'applications Java EE (4 jours)
- Ouvrages
 - **Apache Tomcat 8 - Guide d'administration du serveur Java EE 7 sous Windows et Linux**
 - Editions ENI – Collection « Expert IT »

