

Cahier de TP

« Administration Tomcat

Pré-requis :

- Bonne connexion Internet
- Système d'exploitation recommandé : Linux, MacOS, Windows 10
- JDK11+
- Postgres ou Docker

TP1 : Installation Tomcat

Objectifs du TP

- Mettre en place la distribution
- Écrire un script de démarrage

Mettre en place la distribution

Vérifier la version de Java :

```
$java -version
```

Système Linux

Récupérer la distribution de Tomcat

La décompresser dans un répertoire de travail

Vérifier les droits d'exécution sur les scripts présents dans `$catalina_home/bin/*.sh`

Système Windows

Récupérer soit une archive soit l'installateur

Démarrer le serveur et accéder à l'URL <http://localhost:8080>

Script de démarrage

Le script doit apporter les fonctionnalités suivantes par rapport aux scripts de la distribution de tomcat :

- Positionner les options de la JVM, en particulier `-Xmx` et `-Xms`
- Offrir une fonctionnalité de `restart`

La logique principale du script pourra s'inspirer de l'extrait suivant :

```
case "$1" in
```

```
start)
    start
    ;;
stop)
    stop
    ;;
restart)
    stop
    start
    ;;
*)
    echo $"Usage: tomcat {start|stop|restart}"
    exit
esac
```

TP2 : Le fichier *server.xml*

Objectifs du TP

- Modifier les ports d'écoutes TCP
- Vérifier la valve générant les logs d'accès
- Créer un autre host

Modifier les ports d'écoute

Modifier le port d'écoute pour la commande de shutdown puis le tester avec un telnet

Modifier le port d'écoute HTTP sur 8000, vérifier avec un navigateur l'accès à *http://localhost:8000*

Vérification de la valve des accès

Vérifier que la valve d'accès est bien active et observer les fichiers journaux créés.

Création d'un nouveau host

- Créer un nouveau host associé à une autre adresse IP que *localhost*
Le nouvel host aura également un répertoire différent pour les applications (webapps2 par exemple)
- Essayer d'accéder au host
- Copier dans webapps2 l'application ROOT
- Accéder au nouvel host

TP3 : Mise en place d'une data source

Objectifs du TP

- Utiliser l'annuaire JNDI global
- Mettre en place une source de données PostgreSQL
- Utiliser *tomcat-jdbc-pool*

Installation Postgres

Installer une base Postgres, vous pouvez également utiliser le fichier docker-compose fourni

Création de la base de données

Créer une base de donnée *tomcat*

Initialiser la base avec le script fourni

Installation du driver JDBC

Récupérer le driver jdbc fourni et l'installer dans *\$CATALINA_HOME/lib*

Inscription d'une ressource globale

Inscrire la source de données dans l'annuaire JNDI global sous le nom de ***jdbc/TomcatDB***

Redémarrer le serveur et s'assurer que la configuration est correcte

Installation de l'application web

Reprendre l'application web *testDbApp* sous forme de répertoire et l'installer dans le répertoire ***webapps***.

Vérifier le nom de la datasource qu'utilise cette application.

Modifier *META-INF/context.xml* afin de lier la ressource locale à la ressource globale

Tester votre mise en place en appelant l'URL <http://localhost:8080/testDBApp/testDB.jsp>

TP4 : *Realms*

Objectifs du TP

- Comprendre la sécurisation d'une application web
- Mettre en place un `DataSourceRealm`

Accès à l'application Manager

Mettre à jour la configuration par défaut afin que l'utilisateur ***tomcat/secret*** puisse accéder à l'application Manager.

Trouver le rôle requis pour accéder à l'application

Trouver le fichier realms utilisé, le modifier en conséquence

Tester l'accès à *http://localhost:8080/manager*

Mise en place d'un DataSourceRealm

Déclarer un *DataSourceRealm* utilisant la base de données installée dans lors du TP précédent.

Utiliser ce Realm pour sécuriser, l'accès à l'application Manager

Encoder les mots de passe

TP5 : Persistance *des session*

Objectifs du TP

- Observer la sérialisation des sessions sur le disque

Accès à l'application Manager

- Utiliser l'application exemple *servlets-examples*
- Accéder à l'exemple illustrant les sessions :
<http://localhost:8080/examples/servlets/servlet/SessionExample>
- Positionner plusieurs attributs de session
- Arrêter le serveur
- Rechercher le fichier ***SESSIONS.ser***
- Redémarrer le serveur
- Vérifier la présence des attributs de session

TP6 : *Déploiement d'application*

Objectifs du TP

- Déploiement dynamique avec l'*autoDeploy*
- Déploiement manuel avec l'application Manager
- Automatisation du déploiement par scripts *ant*
- Automatisation du déploiement à partir de fichiers sources

6.1 *Déploiement dynamique*

- Démarrer le serveur
- Déplacer le répertoire *webapps/examples* dans un répertoire de travail
- Observer le fichier *catalina.out*
`tail -f $catalina_home/logs/catalina.out`
Essayer d'accéder à l'application
- Replacer le répertoire dans *webapps*
- Accéder à l'application
- Déplacer à nouveau le répertoire *webapps/examples* dans un répertoire de travail
- Se placer dans le *\$tmp/examples* et créer un *.war* :
`jar -cvf ../examples.war .`
- Copier le fichier *examples.war* généré dans le répertoire *webapps*, observer le contenu du répertoire
- Accéder à l'application
- Modifier le paramètre *unpackWARs* du serveur
- Supprimer l'application exemples de *webapps*
- Redéployer l'archive
- Accéder à l'application
- Modifier le paramètre *autoDeploy* et vérifier que le déploiement dynamique est désactivé

6.2 *Déploiement avec le Manager*

- Rester en mode *autoDeploy=false*
- Accéder à l'application Manager et retirer l'application *examples*
Observer le répertoire *webapps*
- Déployer en effectuant un upload de *examples.war*
- Supprimer à nouveau l'application
- Déployer en utilisant une référence au *.war* ou au répertoire local au serveur

6.3 Automatisation du déploiement par script ant

Installation de ant

Récupérer une distribution de *ant*

Positionner la variable ANT_HOME

et ajouter ANT_HOME/bin dans la variable PATH.

Tester la commande *ant -h*

Ajout de la librairie additionnelle de tomcat

Copier la librairie *\$CATALINA_HOME/lib/catalina-ant.jar* dans *\$ANT_HOME/lib*

Adaptation du script fourni

Récupérer les fichiers *build.xml* et *build.properties* et les placer dans votre répertoire de travail
\$tmp

Adapter les propriétés Ant à votre environnement

Test du script

Exécuter :

ant usage

Exécuter d'autres target en particulier les targets de déploiement et repli

6.4 Déploiement avec ClientDeployer

- Installation du Client Deployer
Télécharger la distribution et la décompresser dans un répertoire de travail
- Récupérer l'application source fournie et la dézipper dans un répertoire de travail
- Créer un fichier *deployer.properties* spécifiant les propriétés propres à votre environnement
- Tester le build et le déploiement de l'application

TP7 : Configuration fichiers journaux

Objectifs du TP

- Comprendre le fichier de configuration JULI

Configuration JULI

- Supprimer le handler `1AsyncFileHandler`
- Afficher le niveau de trace à FINE pour le package `org.apache.catalina`
- Observer `catalina.out`
- Limiter à INFO le package `org.apache.catalina.util.LifecycleBase`
- Créer un nouveau `FileHandler` pour l'application `examples`. Le niveau du handler sera INFO.
Si possible effectuer la configuration dans le fichier `WEB-INF/classes/logging.properties`
- Accéder à l'application et observer les logs
`tail -f $logfile`

TP8 : *Monitoring JMX*

Objectifs du TP

- Observer les attributs des Mbeans Tomcat
- Activer le monitoring distant

Connexion local avec Jconsole

- Démarrer tomcat
- Démarrer la JConsole (*jconsole*)
- Observer l'utilisation mémoire
- Observer les Mbeans sous catalina et modifier dynamiquement certains attributs

Activation du monitoring distant

- Modifier le script de démarrage pour ajouter la propriété `com.sun.management.jmxremote.port`
- Désactiver SSL mais pas l'authentification
- Créer les 2 fichiers contrôlant les accès *jmxremote.password* et *jmxremote.access*
- Démarrer tomcat
- Connectez vous en mode distant avec la JConsole

TP9 : Intégration *Apache* comme *Proxy*

Objectifs du TP

- Voir les différentes alternatives d'intégration entre Apache et Tomcat en Intégrer Apache comme proxy

9.1 HTTP Proxy

Configuration Apache

- Charger les modules *mod_proxy* et *mod_proxy_http*
- Définir le mapping de l'URL */examples* vers Tomcat

Configuration Tomcat

- Ajouter un connecteur http utilisant Apache comme proxy
- Redémarrer les 2 serveurs
- Tester l'url <http://localhost/examples>

9.2 AJP Proxy

Configuration Apache

- Charger les modules *mod_proxy* et *mod_proxy_ajp*
- Définir le mapping de l'URL */examples* vers Tomcat

Configuration Tomcat

- Ajouter un connecteur Ajp utilisant Apache comme proxy
- Redémarrer les 2 serveurs
- Tester l'url <http://localhost/examples>

9.3 Intégration avec *mod_jk*

Configuration Apache

- Récupérer le connecteur tomcat

- Mettre au point un fichier ***workers.properties***
- Charger les modules ***mod_jk***
- Configurer ***mod_jk*** et associer ***/examples*** à un nœud tomcat

Configuration Tomcat

- S'assurer de la présence du connecteur AJP
- Redémarrer les 2 serveurs
- Tester l'url <http://localhost/examples>

TP11 : *Architecture en cluster*

Objectifs du TP

- Intégrer Apache comme répartiteur de charge
- Effectuer une réplication de session

Autoriser plusieurs instances tomcat

- Recopier les répertoires *conf*, *log*, *webapps* dans un autre répertoire : *\$tomcat2*
- Dans *\$tomcat2/conf/server.xml*, Modifier les ports utilisés pour éviter les conflits de port
- Modifier les scripts de démarrage pour utiliser la variable *CATALINA_BASE*
- Tester le démarrage des deux serveurs

Configuration Apache

- Créer un fichiers *workers.properties* utilisant les 2 nœuds et un load balancer, utiliser la sticky session

Configuration Tomcat

- Configurer Tomcat en ajoutant l'attribut **jvmRoute**
- Redémarrer les 2 serveurs
- Tester l'url <http://localhost/examples> avec 2 navigateurs de préférence
- Aller dans l'exemple sur la session utilisateur et observer le cookie JSESSIONID
- Arrêter un serveur et observer le comportement sur les 2 navigateurs
- Redémarrer le serveur arrêté et relancer les navigateurs afin de vérifier qu'Apache route vers le nœud redémarré

Réplication de session

- Décommenter la configuration cluster dans les 2 noeuds
- Reconstruire l'application *examples* avec l'attribut `<distributable/>` dans *web.xml*
- Déployer l'application *distributable* sur les 2 serveurs, puis les démarrer
- Accéder à l'application testant les sessions, repérer le nœud servant la requête
- Stocker une valeur dans la session puis arrêter le nœud correspondant
- Saisir une seconde valeur et s'assurer que la première valeur saisie n'est pas perdue

TP12 : *Sécurisation Tomcat*

- Modifier la commande de shutdown
- Restreindre les connecteurs inutilisés
- Décommenter le Security Lifecycle Listener
- Se protéger des attaques XSS
- Mettre un filtre sur IP