Version : **2023.01**.

Dernière mise-à-jour : 2023/10/07 11:45

SER505 - Sécurité du serveur Tomcat

Contenu du Module

    - 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/tomcat10/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 10 peut utiliser les six Realms suivants :

- User Database 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** :

```
[root@centos8 work]# 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>
```

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.LockOutRealm">
        <!-- This Realm uses the UserDatabase configured in the global JNDI
             resources under the key "UserDatabase".  Any edits
             that are performed against this UserDatabase are immediately
             available for use by the Realm.  -->
        <Realm className="org.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-512**, soit l'algorithme **SHA-256** soit l'algorithme **MD5** :

```
[root@centos8 work]# cd $CATALINA_HOME/bin

[root@centos8 bin]# ./digest.sh -a SHA-256 -h org.apache.catalina.realm.MessageDigestCredentialHandler fenestros
fenestros:f13c89ed8da3d2674c1937503b73fb15cd061751ddbefdb12c337cf0a67c0b0c$1$ad18b00f8856db9fa0396a5448fa022ed2b7
c367faf113e209bb68e16cbffbce
```

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@centos8 bin]# vi $CATALINA_HOME/conf/tomcat-users.xml
[root@centos8 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="f13c89ed8da3d2674c1937503b73fb15cd061751ddbefdb12c337cf0a67c0b0c$1$ad18b00f8856db9fa0396a5448fa022ed2b7
c367faf113e209bb68e16cbffbce" 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** en y ajoutant un **CredentialHandler** :

```
[root@centos8 bin]# vi $CATALINA_HOME/conf/server.xml

[root@centos8 bin]# cat $CATALINA_HOME/conf/server.xml
...
      <Realm className="org.apache.catalina.realm.LockOutRealm">
        <!-- This Realm uses the UserDatabase configured in the global JNDI
             resources under the key "UserDatabase".  Any edits
             that are performed against this UserDatabase are immediately
             available for use by the Realm.  -->
        <Realm className="org.apache.catalina.realm.UserDatabaseRealm"
               resourceName="UserDatabase">
        <CredentialHandler className="org.apache.catalina.realm.MessageDigestCredentialHandler"
algorithm="SHA-256"/>
```

```
      </Realm>
      </Realm>
...
```

Redémarrez le serveur Tomcat :

```
[root@centos8 bin]# systemctl restart tomcat
[root@centos8 bin]# systemctl status tomcat
● tomcat.service - Apache Tomcat Web Application Container
   Loaded: loaded (/etc/systemd/system/tomcat.service; disabled; vendor preset: disabled)
   Active: active (running) since Thu 2023-10-05 03:00:04 EDT; 7s ago
  Process: 85808 ExecStop=/bin/kill -15 $MAINPID (code=exited, status=0/SUCCESS)
  Process: 85817 ExecStart=/usr/tomcat10/bin/startup.sh (code=exited, status=0/SUCCESS)
 Main PID: 85828 (java)
    Tasks: 50 (limit: 100949)
   Memory: 351.1M
   CGroup: /system.slice/tomcat.service
           └─85828 /usr/lib/jvm/jre-1.8.0-openjdk-1.8.0.312.b07-2.el8_5.x86_64/bin/java -
Djava.util.logging.config.file=/usr/tomcat10//conf/logging.properties -
Djava.util.logging.manager=org.apache.juli.ClassL>

Oct 05 03:00:04 centos8.ittraining.loc systemd[1]: Starting Apache Tomcat Web Application Container...
Oct 05 03:00:04 centos8.ittraining.loc startup.sh[85817]: Existing PID file found during start.
Oct 05 03:00:04 centos8.ittraining.loc startup.sh[85817]: Removing/clearing stale PID file.
Oct 05 03:00:04 centos8.ittraining.loc startup.sh[85817]: Tomcat started.
Oct 05 03:00:04 centos8.ittraining.loc systemd[1]: Started Apache Tomcat Web Application Container.
```

Testez votre connexion :

```
[root@centos8 bin]# lynx --dump -auth admin:fenestros "http://www.ittraining.loc:8080/manager/text/serverinfo"
OK - Server info
Tomcat Version: [Apache Tomcat/10.0.27]
OS Name: [Linux]
OS Version: [4.18.0-305.7.1.el8_4.x86_64]
```

```
OS Architecture: [amd64]
JVM Version: [1.8.0_312-b07]
JVM Vendor: [Red Hat, Inc.]
```

DataSource Realm

Ce type de Realm utilise la classe Java **org.apache.catalina.realm.DataSourceRealm**. Les informations sont stockées dans une base de données. Le Realm DataSource peut utiliser un **pool** de connexions qui augmente la performance.

Commencez par créer la base de données **auth_tomcat** :

```
[root@centos8 bin]# mysql -uroot -p
Enter password: fenestros
Welcome to the MariaDB monitor.  Commands end with ; or \g.
Your MariaDB connection id is 37
Server version: 10.3.28-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)]> CREATE DATABASE auth_tomcat;
Query OK, 1 row affected (0.000 sec)

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

```
+--------------------+
5 rows in set (0.001 sec)

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

Les informations sont stockées dans deux tables de la base de données - **users** et **roles** :

```
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@centos8 bin]# cd ~
[root@centos8 ~]# vi auth_tomcat
[root@centos8 ~]# cat 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 les tables :

```
[root@centos8 ~]# mysql -u root -p < auth_tomcat
Enter password: fenestros
```

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

```
[root@centos8 ~]# mysql -u root -p
Enter password: fenestros
Welcome to the MariaDB monitor.  Commands end with ; or \g.
Your MariaDB connection id is 39
Server version: 10.3.28-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 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
(0.073 sec)

MariaDB [auth_tomcat]> INSERT INTO `auth_tomcat`.`roles` VALUES('admin1','manager-script');
Query OK, 1 row affected (0.134 sec)

MariaDB [auth_tomcat]> GRANT SELECT ON auth_tomcat.* TO 'tomcat'@'localhost' IDENTIFIED BY 'tomcat';
Query OK, 0 rows affected (0.001 sec)
```

```
MariaDB [auth_tomcat]> FLUSH PRIVILEGES;
Query OK, 0 rows affected (0.002 sec)

MariaDB [auth_tomcat]> SET PASSWORD FOR 'tomcat'@'localhost' = PASSWORD('secret');
Query OK, 0 rows affected (0.000 sec)

MariaDB [auth_tomcat]> exit
Bye
```

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

```
[root@centos8 bin]# vi $CATALINA_HOME/conf/server.xml

[root@centos8 bin]# cat $CATALINA_HOME/conf/server.xml
...
  <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.cj.jdbc.Driver"
                 url="jdbc:mysql://localhost:3306/auth_tomcat"
            username="tomcat"
            password="secret" />

  </GlobalNamingResources>
```

```
...
```

Commentez ensuite le Realm **UserDatabaseRealm** et ajoutez le Realm **DataSourceRealm** :

```
[root@centos8 bin]# vi $CATALINA_HOME/conf/server.xml

[root@centos8 bin]# cat $CATALINA_HOME/conf/server.xml
...
      <Realm className="org.apache.catalina.realm.LockOutRealm">
        <!-- This Realm uses the UserDatabase configured in the global JNDI
             resources under the key "UserDatabase".  Any edits
             that are performed against this UserDatabase are immediately
             available for use by the Realm.  -->
        <!-- <Realm className="org.apache.catalina.realm.UserDatabaseRealm"
               resourceName="UserDatabase">
        <CredentialHandler className="org.apache.catalina.realm.MessageDigestCredentialHandler"
algorithm="SHA-256"/>
      </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@centos8 bin]# systemctl restart tomcat
[root@centos8 bin]# systemctl status tomcat
● tomcat.service - Apache Tomcat Web Application Container
   Loaded: loaded (/etc/systemd/system/tomcat.service; disabled; vendor preset: disabled)
   Active: active (running) since Thu 2023-10-05 05:04:39 EDT; 4s ago
```

```
  Process: 88183 ExecStop=/bin/kill -15 $MAINPID (code=exited, status=0/SUCCESS)
  Process: 88191 ExecStart=/usr/tomcat10/bin/startup.sh (code=exited, status=0/SUCCESS)
 Main PID: 88201 (java)
    Tasks: 50 (limit: 100949)
   Memory: 348.2M
   CGroup: /system.slice/tomcat.service
           └─88201 /usr/lib/jvm/jre-1.8.0-openjdk-1.8.0.312.b07-2.el8_5.x86_64/bin/java -
Djava.util.logging.config.file=/usr/tomcat10//conf/logging.properties -
Djava.util.logging.manager=org.apache.juli.ClassL>

Oct 05 05:04:39 centos8.ittraining.loc systemd[1]: Starting Apache Tomcat Web Application Container...
Oct 05 05:04:39 centos8.ittraining.loc startup.sh[88191]: Existing PID file found during start.
Oct 05 05:04:39 centos8.ittraining.loc startup.sh[88191]: Removing/clearing stale PID file.
Oct 05 05:04:39 centos8.ittraining.loc startup.sh[88191]: Tomcat started.
Oct 05 05:04:39 centos8.ittraining.loc systemd[1]: Started Apache Tomcat Web Application Container..
```

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

```
[root@centos8 bin]# lynx --dump -auth admin1:fenestros "http://www.ittraining.loc:8080/manager/text/serverinfo"
OK - Server info
Tomcat Version: [Apache Tomcat/10.0.27]
OS Name: [Linux]
OS Version: [4.18.0-305.7.1.el8_4.x86_64]
OS Architecture: [amd64]
JVM Version: [1.8.0_312-b07]
JVM Vendor: [Red Hat, Inc.]
```

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@centos8 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>] any libldap ldap.conf options, plus
             ldif_wrap=<width> (in columns, or "no" for no wrapping)
             nettimeout=<timeout> (in seconds, or "none" or "max")
  -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** :

```
[root@centos8 bin]# vi /etc/openldap/slapd.conf

[root@centos8 bin]# cat /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@centos8 bin]# rm -Rf /etc/openldap/slapd.d/*

[root@centos8 bin]# rm -f /var/lib/ldap/alock

[root@centos8 bin]# rm -f /var/lib/ldap/__db.00?
```

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

```
[root@centos8 bin]# cp /usr/share/openldap-servers/DB_CONFIG.example /var/lib/ldap/DB_CONFIG
```

Copiez le fichier **/usr/share/openldap-servers/slapd.ldif** vers **/etc/openldap/** :

```
[root@centos8 bin]# cp /usr/share/openldap-servers/slapd.ldif /etc/openldap/
```

Générez maintenant un mot de passe pour l'administrateur d'OpenLDAP :

```
[root@centos8 tmp]# slappasswd
New password: fenestros
Re-enter new password: fenestros
{SSHA}dAVyaIZX7WT4DBu6/8yHwQ12+YoTt5os
```

La commande slappasswd prend les options suivantes :

```
[root@centos8 tmp]# slappasswd --help
slappasswd: invalid option -- '-'
Usage: slappasswd [options]
  -c format     crypt(3) salt format
  -g            generate random password
  -h hash       password scheme
  -n            omit trailing newline
  -o <opt>[=val] specify an option with a(n optional) value
        module-path=<pathspec>
        module-load=<filename>
  -s secret     new password
  -u            generate RFC2307 values (default)
  -v            increase verbosity
  -T file       read file for new password
```

Il convient ensuite de modifier le fichier **/etc/openldap/slapd.ldif** en y ajoutant la ligne **olcRootPW: {SSHA}dAVyaIZX7WT4DBu6/8yHwQ12+YoTt5os**. Les directives **olcSuffix: dc=my-domain,dc=com** et **olcRootDN: cn=Manager,dc=my-domain,dc=com** doivent être modifiées pour votre système ainsi :

```
...
olcSuffix: o=ittraining.loc
olcRootDN: cn=Manager,o=ittraining.loc
...
```

Vous obtiendrez :

```
[root@centos8 tmp]# vi /etc/openldap/slapd.ldif

[root@centos8 tmp]# cat /etc/openldap/slapd.ldif
...
#
# Backend database definitions
#

dn: olcDatabase=mdb,cn=config
objectClass: olcDatabaseConfig
objectClass: olcMdbConfig
olcDatabase: mdb
olcSuffix: o=ittraining.loc
olcRootDN: cn=Manager,o=ittraining.loc
olcRootPW: {SSHA}dAVyaIZX7WT4DBu6/8yHwQ12+YoTt5os
olcDbDirectory: /var/lib/ldap
olcDbIndex: objectClass eq,pres
olcDbIndex: ou,cn,mail,surname,givenname eq,pres,sub
```

> **Important** - La directive **olcSuffix** indique la racine de l'arbre qui est détenu dans la base de données. La directive **olcRootDN** indique les coordonnées de connexion de l'administrateur de cet arbre. N'utilisez pas cn=**root**.

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

```
[root@centos8 bin]# slapadd -F /etc/openldap/slapd.d/ -n 0 -l /etc/openldap/slapd.ldif
_################### 100.00% eta   none elapsed            none fast!
Closing DB...
```

Vérifiez que l'arborescence initiale soit créée :

```
[root@centos8 bin]# ls -l /etc/openldap/slapd.d
total 4
drwxr-x--- 3 ldap ldap 182 Oct  5 08:49 'cn=config'
-rw------- 1 ldap ldap 366 Oct  5 08:49 'cn=config.ldif'
```

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

```
[root@centos8 bin]# chown -R ldap:ldap /etc/openldap/slapd.d

[root@centos8 bin]# chmod -R u+rwX /etc/openldap/slapd.d
```

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

```
[root@centos8 bin]# chown -R ldap:ldap /var/lib/ldap/ /etc/openldap/slapd.conf
```

Vous pouvez maintenant tester votre configuration :

```
[root@centos8 bin]# slaptest -u
config file testing succeeded
```

Démarrez le service slapd :

```
[root@centos8 bin]# systemctl start slapd

[root@centos8 bin]# systemctl status slapd
● slapd.service - OpenLDAP Server Daemon
   Loaded: loaded (/usr/lib/systemd/system/slapd.service; enabled; vendor preset: disabled)
   Active: active (running) since Thu 2023-10-05 08:51:26 EDT; 7s ago
     Docs: man:slapd
           man:slapd-config
           man:slapd-hdb
           man:slapd-mdb
           file:///usr/share/doc/openldap-servers/guide.html
```

```
  Process: 90402 ExecStart=/usr/sbin/slapd -u ldap -h ldap:/// ldaps:/// ldapi:/// (code=exited,
status=0/SUCCESS)
  Process: 90388 ExecStartPre=/usr/libexec/openldap/check-config.sh (code=exited, status=0/SUCCESS)
 Main PID: 90403 (slapd)
    Tasks: 2 (limit: 100949)
   Memory: 3.0M
   CGroup: /system.slice/slapd.service
           └─90403 /usr/sbin/slapd -u ldap -h ldap:/// ldaps:/// ldapi:///

Oct 05 08:51:26 centos8.ittraining.loc systemd[1]: Starting OpenLDAP Server Daemon...
Oct 05 08:51:26 centos8.ittraining.loc runuser[90391]: pam_unix(runuser:session): session opened for user ldap by
(uid=0)
Oct 05 08:51:26 centos8.ittraining.loc runuser[90391]: pam_unix(runuser:session): session closed for user ldap
Oct 05 08:51:26 centos8.ittraining.loc slapd[90402]: @(#) $OpenLDAP: slapd 2.4.46 (Aug 10 2021 05:11:20) $
mockbuild@x86-02.mbox.centos.org:/builddir/build/BUILD/openldap-2.4.46/openldap-2.4.46/servers/slapd
Oct 05 08:51:26 centos8.ittraining.loc slapd[90403]: slapd starting
Oct 05 08:51:26 centos8.ittraining.loc systemd[1]: Started OpenLDAP Server Daemon.
```

Créez maintenant notre fichier LDIF dans le répertoire /tmp :

```
[root@centos8 bin]# cd /tmp

[root@centos8 tmp]# vi setup.ldif

[root@centos8 tmp]# cat setup.ldif
dn: o=ittraining.loc
objectClass: dcObject
objectClass: organization
dc: ittraining
o: ittraining.loc
description: Exemple

dn: ou=utilisateurs,o=ittraining.loc
objectClass: organizationalUnit
```

```
objectClass: top
ou: utilisateurs

dn: cn=admin2,ou=utilisateurs,o=ittraining.loc
objectClass: person
objectClass: top
cn: admin2
sn: admin2
userPassword: fenestros

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

dn: cn=manager-script,ou=roles,o=ittraining.loc
objectClass: groupOfUniqueNames
objectClass: top
cn: manager-script
uniqueMember: cn=admin2,ou=utilisateurs,o=ittraining.loc
```

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

```
[root@centos8 tmp]# ldapadd -f setup.ldif -x -D "cn=Manager,o=ittraining.loc" -w fenestros
adding new entry "o=ittraining.loc"

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

adding new entry "cn=admin2,ou=utilisateurs,o=ittraining.loc"

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

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

L'arborescence LDAP est la suivante :

```
localhost
        |
    o=ittraining.loc
        |
        ou=roles
        |    |
        |    cn=manager-script
        |
        ou=users
            |
            cn=admin2
```

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

```
[root@centos8 tmp]# vi $CATALINA_HOME/conf/server.xml

[root@centos8 tmp]# cat $CATALINA_HOME/conf/server.xml
...
      <!-- Use the LockOutRealm to prevent attempts to guess user passwords
           via a brute-force attack -->
      <Realm className="org.apache.catalina.realm.LockOutRealm">
        <!-- This Realm uses the UserDatabase configured in the global JNDI
             resources under the key "UserDatabase".  Any edits
             that are performed against this UserDatabase are immediately
             available for use by the Realm.  -->
        <!-- <Realm className="org.apache.catalina.realm.UserDatabaseRealm"
               resourceName="UserDatabase">
        <CredentialHandler className="org.apache.catalina.realm.MessageDigestCredentialHandler"
algorithm="SHA-256"/>
      </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  className="org.apache.catalina.realm.JNDIRealm"
          connectionURL="ldap://localhost:389"
         connectionName="cn=Manager,o=ittraining.loc"
     connectionPassword="fenestros"
               roleBase="ou=roles,o=ittraining.loc"
               roleName="cn"
             roleSearch="(uniqueMember={0})"
           userPassword="userPassword"
            userPattern="cn={0},ou=utilisateurs,o=ittraining.loc" />

      </Realm>

      <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 ensuite le service tomcat :

```
[root@centos8 tmp]# systemctl restart tomcat

[root@centos8 tmp]# systemctl status tomcat
● tomcat.service - Apache Tomcat Web Application Container
   Loaded: loaded (/etc/systemd/system/tomcat.service; disabled; vendor preset: disabled)
   Active: active (running) since Thu 2023-10-05 09:57:34 EDT; 19s ago
  Process: 91072 ExecStop=/bin/kill -15 $MAINPID (code=exited, status=0/SUCCESS)
  Process: 91213 ExecStart=/usr/tomcat10/bin/startup.sh (code=exited, status=0/SUCCESS)
 Main PID: 91223 (java)
    Tasks: 51 (limit: 100949)
   Memory: 356.7M
   CGroup: /system.slice/tomcat.service
           └─91223 /usr/lib/jvm/jre-1.8.0-openjdk-1.8.0.312.b07-2.el8_5.x86_64/bin/java -
Djava.util.logging.config.file=/usr/tomcat10//conf/logging.properties -
Djava.util.logging.manager=org.apache.juli.ClassL>

Oct 05 09:57:34 centos8.ittraining.loc systemd[1]: Starting Apache Tomcat Web Application Container...
Oct 05 09:57:34 centos8.ittraining.loc startup.sh[91213]: Existing PID file found during start.
Oct 05 09:57:34 centos8.ittraining.loc startup.sh[91213]: Removing/clearing stale PID file.
Oct 05 09:57:34 centos8.ittraining.loc startup.sh[91213]: Tomcat started.
Oct 05 09:57:34 centos8.ittraining.loc systemd[1]: Started Apache Tomcat Web Application Container.
```

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

```
[root@centos8 tmp]# lynx --dump -auth admin2:fenestros "http://www.ittraining.loc:8080/manager/text/serverinfo"
OK - Server info
Tomcat Version: [Apache Tomcat/10.0.27]
OS Name: [Linux]
OS Version: [4.18.0-305.7.1.el8_4.x86_64]
OS Architecture: [amd64]
JVM Version: [1.8.0_312-b07]
JVM Vendor: [Red Hat, Inc.]
```

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>
```

Important : Pour plus d'information concernant les Realms, consultez le **manuel** de Tomcat.

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> du fichier **$CATALINA_HOME/conf/server.xml**.

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@centos8 tmp]# 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@centos8 tmp]# keytool -genkey -alias tomcat -keyalg RSA -keystore
$CATALINA_HOME/security/www_ittraining_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]:  ITTRAINING
What is the name of your organization?
  [Unknown]:  TEAM ITTRAINING
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=ITTRAINING, O=TEAM ITTRAINING, L=ADDLESTON, ST=SURREY, C=GB correct?
  [no]:  YES

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

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/tomcat10/security/www_ittraining_loc.keystore -
destkeystore /usr/tomcat10/security/www_ittraining_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@centos8 tmp]# keytool -certreq -keyalg RSA -alias tomcat -file www_ittraining_loc.csr -keystore
$CATALINA_HOME/security/www_ittraining_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/tomcat10/security/www_ittraining_loc.keystore -
destkeystore /usr/tomcat10/security/www_ittraining_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>
```

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@centos8 tmp]# keytool -selfcert -alias tomcat -keypass fenestros -keystore
$CATALINA_HOME/security/www_ittraining_loc.keystore -dname "CN=HUGH NORRIS, OU=ITTRAINING, O=TEAM ITTRAINING,
L=ADDLESTON, ST=SURREY, C=GB"
Enter keystore password:

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/tomcat10/security/www_ittraining_loc.keystore -
destkeystore /usr/tomcat10/security/www_ittraining_loc.keystore -deststoretype pkcs12".
```

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

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

```
[root@centos8 tmp]# vi $CATALINA_HOME/conf/server.xml

[root@centos8 tmp]# cat $CATALINA_HOME/conf/server.xml
...
    <!--
    <Connector port="8443" protocol="org.apache.coyote.http11.Http11NioProtocol"
               maxThreads="150" SSLEnabled="true">
        <UpgradeProtocol className="org.apache.coyote.http2.Http2Protocol" />
        <SSLHostConfig>
            <Certificate certificateKeystoreFile="conf/localhost-rsa.jks"
                         type="RSA" />
        </SSLHostConfig>
    </Connector>
    -->

    <Connector port="8443" protocol="org.apache.coyote.http11.Http11NioProtocol"
               maxThreads="150" SSLEnabled="true">
        <UpgradeProtocol className="org.apache.coyote.http2.Http2Protocol" />
        <SSLHostConfig>
            <Certificate
            certificateKeystoreFile="/usr/tomcat10/security/www_ittraining_loc.keystore"
            certificateKeystorePassword="fenestros"
            type="RSA"
            />
        </SSLHostConfig>
    </Connector>

    <!-- Define an AJP 1.3 Connector on port 8009 -->
    <Connector protocol="AJP/1.3"
               address="127.0.0.1"
               port="8009"
               redirectPort="8443" secretRequired="false" />
...
```

Redémarrez le serveur Tomcat :

```
[root@centos8 tmp]# systemctl restart tomcat
[root@centos8 tmp]# systemctl status tomcat
● tomcat.service - Apache Tomcat Web Application Container
   Loaded: loaded (/etc/systemd/system/tomcat.service; disabled; vendor preset: disabled)
   Active: active (running) since Thu 2023-10-05 10:21:29 EDT; 7s ago
  Process: 91573 ExecStop=/bin/kill -15 $MAINPID (code=exited, status=0/SUCCESS)
  Process: 91582 ExecStart=/usr/tomcat10/bin/startup.sh (code=exited, status=0/SUCCESS)
 Main PID: 91594 (java)
    Tasks: 51 (limit: 100949)
   Memory: 343.4M
   CGroup: /system.slice/tomcat.service
           └─91594 /usr/lib/jvm/jre-1.8.0-openjdk-1.8.0.312.b07-2.el8_5.x86_64/bin/java -
Djava.util.logging.config.file=/usr/tomcat10//conf/logging.properties -
Djava.util.logging.manager=org.apache.juli.ClassL>

Oct 05 10:21:29 centos8.ittraining.loc systemd[1]: Starting Apache Tomcat Web Application Container...
Oct 05 10:21:29 centos8.ittraining.loc startup.sh[91582]: Existing PID file found during start.
Oct 05 10:21:29 centos8.ittraining.loc startup.sh[91582]: Removing/clearing stale PID file.
Oct 05 10:21:29 centos8.ittraining.loc startup.sh[91582]: Tomcat started.
Oct 05 10:21:29 centos8.ittraining.loc systemd[1]: Started Apache Tomcat Web Application Container.
```

Vérifiez maintenant que vous pouvez vous connecter avec le compte **admin2** sur le port **8443** :

```
[root@centos8 tmp]# curl -k --user admin2:fenestros "https://www.ittraining.loc:8443/manager/text/serverinfo"
OK - Server info
Tomcat Version: [Apache Tomcat/10.0.27]
OS Name: [Linux]
OS Version: [4.18.0-305.7.1.el8_4.x86_64]
OS Architecture: [amd64]
JVM Version: [1.8.0_312-b07]
JVM Vendor: [Red Hat, Inc.]
```

Configurer Apache

Installation de SSL

Installez le module **mod_ssl** pour Apache :

```
[root@centos8 tmp]# rpm -qa | grep ssl
openssl-pkcs11-0.4.10-2.el8.x86_64
openssl-devel-1.1.1g-15.el8_3.x86_64
openssl-1.1.1g-15.el8_3.x86_64
apr-util-openssl-1.6.1-6.el8.x86_64
xmlsec1-openssl-1.2.25-4.el8.x86_64
openssl-libs-1.1.1g-15.el8_3.x86_64

[root@centos8 tmp]# dnf install mod_ssl
Last metadata expiration check: 2:31:44 ago on Thu 05 Oct 2023 08:36:23 EDT.
Dependencies resolved.
================================================================================
 Package                    Architecture                Version
Repository                    Size
================================================================================
Installing:
 mod_ssl                    x86_64
1:2.4.37-43.module_el8.5.0+1022+b541f3b1                    appstream
136 k

Transaction Summary
================================================================================
Install  1 Package
```

```
Total download size: 136 k
Installed size: 266 k
Is this ok [y/N]: y
```

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@centos8 tmp]# cd /root ; openssl genrsa -out www.ittraining.loc.key 2048
Generating RSA private key, 1024 bit long modulus (2 primes)
.......+++++
..................+++++
e is 65537 (0x010001)
```

Générer maintenant votre CSR :

```
[root@centos8 ~]# openssl req -new -key www.ittraining.loc.key -out www.ittraining.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]:ADDLESTONE
Organization Name (eg, company) [Default Company Ltd]:ITTRAINING
Organizational Unit Name (eg, section) []:TEAM ITTRAINING
```

```
Common Name (eg, your name or your server's hostname) []:www.ittraining.loc
Email Address []:infos@ittraining.loc

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

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@centos8 ~]# openssl x509 -req -days 365 -in www.ittraining.loc.csr -signkey www.ittraining.loc.key -out
www.ittraining.loc.crt
Signature ok
subject=C = GB, ST = SURREY, L = ADDLESTONE, O = ITTRAINING, OU = TEAM ITTRAINING, CN = www.ittraining.loc,
emailAddress = infos@ittraining.loc
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@centos8 ~]# cp /root/www.ittraining.loc.key /etc/pki/tls/private/

[root@centos8 ~]# cp /root/www.ittraining.loc.crt /etc/pki/tls/certs/
```

Mise en place des paramètres de sécurité SSL

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

```
[root@centos8 ~]# 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@centos8 ~]# 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.ittraining.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.ittraining.loc.crt
```

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

Sauvegardez le fichier et re-démarrez le serveur Apache :

```
[root@centos8 ~]# systemctl restart httpd.service
[root@centos8 ~]# 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 Thu 2023-10-05 11:37:56 EDT; 8s ago
     Docs: man:httpd.service(8)
 Main PID: 3207 (httpd)
   Status: "Started, listening on: port 443, port 80"
    Tasks: 213 (limit: 100949)
   Memory: 39.6M
   CGroup: /system.slice/httpd.service
           ├─3207 /usr/sbin/httpd -DFOREGROUND
           ├─3210 /usr/sbin/httpd -DFOREGROUND
           ├─3211 /usr/sbin/httpd -DFOREGROUND
           ├─3212 /usr/sbin/httpd -DFOREGROUND
           └─3213 /usr/sbin/httpd -DFOREGROUND

Oct 05 11:37:56 centos8.ittraining.loc systemd[1]: Starting The Apache HTTP Server...
Oct 05 11:37:56 centos8.ittraining.loc systemd[1]: Started The Apache HTTP Server.
Oct 05 11:37:56 centos8.ittraining.loc httpd[3207]: Server configured, listening on: port 443, port 80
```

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@centos8 ~]# openssl s_client -connect www.ittraining.loc:443
CONNECTED(00000003)
depth=0 C = GB, ST = SURREY, L = ADDLESTONE, O = ITTRAINING, OU = TEAM ITTRAINING, CN = www.ittraining.loc,
emailAddress = infos@ittraining.loc
```

```
verify error:num=18:self signed certificate
verify return:1
depth=0 C = GB, ST = SURREY, L = ADDLESTONE, O = ITTRAINING, OU = TEAM ITTRAINING, CN = www.ittraining.loc,
emailAddress = infos@ittraining.loc
verify return:1
---
Certificate chain
 0 s:C = GB, ST = SURREY, L = ADDLESTONE, O = ITTRAINING, OU = TEAM ITTRAINING, CN = www.ittraining.loc,
emailAddress = infos@ittraining.loc
   i:C = GB, ST = SURREY, L = ADDLESTONE, O = ITTRAINING, OU = TEAM ITTRAINING, CN = www.ittraining.loc,
emailAddress = infos@ittraining.loc
---
Server certificate
-----BEGIN CERTIFICATE-----
MIID0TCCArkCFFLeGM6PP1JjFbcjKPjYv3EWfEKJMA0GCSqGSIb3DQEBCwUAMIGk
MQswCQYDVQQGEwJHQjEPMA0GA1UECAwGU1VSUkVZMRMwEQYDVQQHDApBRERMRVNU
T05FMRMwEQYDVQQKDApJVFRSQUlOSU5HMRgwFgYDVQQLDA9URUFNIElUVFJBSU5J
TkcxGzAZBgNVBAMMEnd3dy5pdHRyYWluaW5nLmxvYzEjMCEGCSqGSIb3DQEJARYU
aW5mb3NAaXR0cmFpbmluZy5sb2MwHhcNMjMxMDA1MTUzNjUzWhcNMjQxMDA0MTUz
NjUzWjCBpDELMAkGA1UEBhMCR0IxDzANBgNVBAgMBlNVUlJFWTETMBEGA1UEBwwK
QURETEVTVE9ORTETMBEGA1UECgwKSVRUUkFJTklORzEYMBYGA1UECwwPVEVBTSBJ
VFRSQUlOSU5HMRswGQYDVQQDDBJ3d3cuaXR0cmFpbmluZy5sb2MxIzAhBgkqhkiG
9w0BCQEWFGluZm9zQGl0dHJhaW5pbmcubG9jMIIBIjANBgkqhkiG9w0BAQEFAAOC
AQ8AMIIBCgKCAQEArf5ui9CzF/wVnx+0XT8s2SO9UJVob8psfK9aY5WqzchZNP5T
DlQY0ys57hOBBrREqa8r61MvdY2f0hF6MhG6IPa06b266QzOCSsRzCr1BsYSR2LB
g/4Avx3DlFGY4Lx7tVkvwiZlVoxShgOgvf20VLeizSKAeSN2LoJ6q9BSaKYmjXpb
zZonns/kUZyiSc9yKKSjIsxnhHLiO6nRDOvXZmv7mORcKsyCAhdHdr33M4gTulG7
yoVXPpGnk/8v3nN4kwRmvHx1SJzMpqhpyFZ2jqug5QAeGPs2rqr7VQayjKXSO+/F
RqrpevzPMoiRbs9Bh9IQJJIMCmqLGn+F7TvAxwIDAQABMA0GCSqGSIb3DQEBCwUA
A4IBAQAY9rXmlliGtXWONSKgIbVlE512joymYscK77bRyeSgkaG5Wo5lZNff2K0z
+BsM13fU1chl83WfaYVQ3+/0jjMR3XoNKU9Oz/kQPJNNGCLOTiBO+PMmLaOJNphT
1axGfgoOtR1HGqty/WpDzYIHzGoZq6j6xNhSkVWvsNg2ockNSaJvrXJO0ZguYudj
/Xh0LPIxwyWOOAU+joeYsanSYtTVYIzPwaighfrROU9iFQ/ld96mMQg3T8dyM+oq
k7QzB+LPTzBwobGq5yVgeqh0Oykwit+05Gg0aJTOG/KJcmoVpCofKJzK4AQ0EomM
```

```
gX1kYEsnI7SgmaAHCCR9vitpg5/W
-----END CERTIFICATE-----
subject=C = GB, ST = SURREY, L = ADDLESTONE, O = ITTRAINING, OU = TEAM ITTRAINING, CN = www.ittraining.loc,
emailAddress = infos@ittraining.loc

issuer=C = GB, ST = SURREY, L = ADDLESTONE, O = ITTRAINING, OU = TEAM ITTRAINING, CN = www.ittraining.loc,
emailAddress = infos@ittraining.loc

---
No client certificate CA names sent
Peer signing digest: SHA256
Peer signature type: RSA-PSS
Server Temp Key: X25519, 253 bits
---
SSL handshake has read 1541 bytes and written 396 bytes
Verification error: self signed certificate
---
New, TLSv1.3, Cipher is TLS_AES_256_GCM_SHA384
Server public key is 2048 bit
Secure Renegotiation IS NOT supported
Compression: NONE
Expansion: NONE
No ALPN negotiated
Early data was not sent
Verify return code: 18 (self signed certificate)
---
---
Post-Handshake New Session Ticket arrived:
SSL-Session:
    Protocol  : TLSv1.3
    Cipher    : TLS_AES_256_GCM_SHA384
    Session-ID: 6DD7C9E55FCD378DAB4976B29C4C2D7C542DB6BD3738926375A8C15EE19CA2FB
    Session-ID-ctx:
    Resumption PSK:
```

```
61F0B32B59E95ED592798EEF7245F3512D7CCE3270E30F6D0A12FB9300B3C21526E0EB8B66D49091510A20151B13B176
    PSK identity: None
    PSK identity hint: None
    SRP username: None
    TLS session ticket lifetime hint: 300 (seconds)
    TLS session ticket:
    0000 - 2c ef f4 b1 99 60 7f 2b-b8 40 f3 09 42 51 92 e6   ,....`.+.@..BQ..
    0010 - 01 73 59 b1 a6 be cc dc-16 36 6c 23 09 4b 26 1f   .sY......6l#.K&.
    0020 - 85 9f 51 2d 66 20 1e b3-d8 42 d2 81 5d 01 f3 37   ..Q-f ...B..]..7
    0030 - 5b ac 4f 14 f0 10 ee 49-a0 87 b5 4d 5b 5b 45 02   [.O....I...M[[E.
    0040 - 2f 29 72 f7 81 c5 c9 7a-4b 42 bb 96 74 b2 60 c7   /)r....zKB..t.`.
    0050 - 6b 47 25 06 04 c5 dc a7-a1 bb 45 39 77 6a 75 90   kG%.......E9wju.
    0060 - c7 20 83 31 21 cb af 0b-51 1a 34 01 5c 9c 79 71   . .1!...Q.4.\.yq
    0070 - 09 52 db a3 cc 3b 42 3d-d7 ba 01 85 7e 65 e7 8b   .R...;B=....~e..
    0080 - b5 c9 77 b9 08 0c 2d c7-21 47 7f 22 ae 6f 18 71   ..w...-.!G.".o.q
    0090 - 8f 5b 97 df 2b ea 30 5f-c9 8d db 96 b9 d2 0e 56   .[..+.0_.......V
    00a0 - be d7 58 3d ee 1b 79 2e-4f b9 77 1e 65 bd 95 b9   ..X=..y.O.w.e...
    00b0 - 99 8b f6 a4 b3 a0 4e 20-c9 c6 f6 18 da a6 66 c6   ......N ......f.
    00c0 - 39 4a 60 83 83 4a c3 24-a2 d9 9b 4a 33 4c 72 34   9J`..J.$...J3Lr4
    00d0 - ce ec 02 4b a8 e4 a4 76-c8 6d fc 00 8c 1b a1 a5   ...K...v.m......
    00e0 - e1 1f 7b 6b 4d d8 a3 41-77 f1 53 8b 22 84 6e eb   ..{kM..Aw.S.".n.
    00f0 - 2f 2e 29 b6 da c1 6b 7a-c0 76 cb ea be f9 ad 19   /.)...kz.v......

    Start Time: 1696520379
    Timeout   : 7200 (sec)
    Verify return code: 18 (self signed certificate)
    Extended master secret: no
    Max Early Data: 0
---
read R BLOCK
---
Post-Handshake New Session Ticket arrived:
SSL-Session:
    Protocol  : TLSv1.3
```

```
    Cipher    : TLS_AES_256_GCM_SHA384
    Session-ID: 3C45CC49320412A018C8BA2425879E379E0E7D4A83B036CFFD9EBCAD0B8CDF7D
    Session-ID-ctx:
    Resumption PSK:
55A93FF2F8185E89F12AAC530C7FB491D7C1F4B19CB125C7AAC41A641CFA25AA21F80BE00FC85A334CC6FFCC912DCC1D
    PSK identity: None
    PSK identity hint: None
    SRP username: None
    TLS session ticket lifetime hint: 300 (seconds)
    TLS session ticket:
    0000 - 2c ef f4 b1 99 60 7f 2b-b8 40 f3 09 42 51 92 e6   ,....`.+.@..BQ..
    0010 - cd 48 14 07 39 6e 1a 12-5f e5 df 5f 2d c9 2b f4   .H..9n.._.._-.+.
    0020 - e8 2d 75 f1 38 de 52 5f-dc 92 b2 1c 98 24 8c 32   .-u.8.R_.....$.2
    0030 - 04 a6 cf 6b dc 15 83 f2-ac a3 8b 24 0a e7 8c 45   ...k.......$...E
    0040 - 37 38 69 ff 4c 12 44 4d-c8 56 ad b2 6c 34 0c 54   78i.L.DM.V..l4.T
    0050 - 38 2f 4c e4 66 2f f8 de-9f c6 fa 44 c5 f5 1b 8e   8/L.f/.....D....
    0060 - a2 77 7c bd 64 59 10 23-41 2b c3 a4 ca cc 31 cd   .w|.dY.#A+....1.
    0070 - 43 5f 44 68 57 b8 bc fd-a8 de 25 3c 8e 63 a6 96   C_DhW.....%<.c..
    0080 - aa 86 42 22 9e f2 93 8d-69 7d e6 2a 77 0b 57 99   ..B"....i}.*w.W.
    0090 - 42 09 6c 3d 1b 4a 96 69-1a 54 71 1b 3e c3 99 4c   B.l=.J.i.Tq.>..L
    00a0 - 19 23 0d 0f ec 10 93 3e-a9 7b 5f 32 75 00 eb d9   .#.....>.{_2u...
    00b0 - ba 19 42 a1 a0 ce ff 60-63 cf 11 9f 68 f0 91 fa   ..B....`c...h...
    00c0 - 51 64 04 ab 2a 26 ab dd-b3 d3 5e 30 c6 f7 c2 7a   Qd..*&....^0...z
    00d0 - 4f b0 be bf e2 5b f4 cc-0c 1a d3 62 5c ea 21 7d   O....[.....b\.!}
    00e0 - 5b 57 af 24 15 fb 93 79-3d 2b 04 76 10 be 91 7f   [W.$...y=+.v....
    00f0 - 29 bf 2b 6f bc 99 61 f9-2a fe d2 c3 41 97 11 6a   ).+o..a.*...A..j

    Start Time: 1696520379
    Timeout   : 7200 (sec)
    Verify return code: 18 (self signed certificate)
    Extended master secret: no
    Max Early Data: 0
---
read R BLOCK
```

```
closed
[root@centos8 ~]#
```

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@centos8 ~]# 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_hcheck_module modules/mod_proxy_hcheck.so
LoadModule proxy_scgi_module modules/mod_proxy_scgi.so
LoadModule proxy_uwsgi_module modules/mod_proxy_uwsgi.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@centos8 ~]# vi /etc/httpd/conf.d/ssl.conf

[root@centos8 ~]# 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.ittraining.loc:8443/docs
    ProxyPassReverse    /docs         http://www.ittraining.loc:8443/docs
</IfModule>

</VirtualHost>
```

Sauvegardez et relancez le service httpd :

```
[root@centos8 ~]# systemctl restart httpd.service

[root@centos8 ~]# 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 Thu 2023-10-05 11:45:48 EDT; 8s ago
     Docs: man:httpd.service(8)
 Main PID: 3578 (httpd)
   Status: "Started, listening on: port 443, port 80"
    Tasks: 213 (limit: 100949)
   Memory: 37.5M
   CGroup: /system.slice/httpd.service
```

```
           ├─3578 /usr/sbin/httpd -DFOREGROUND
           ├─3579 /usr/sbin/httpd -DFOREGROUND
           ├─3580 /usr/sbin/httpd -DFOREGROUND
           ├─3581 /usr/sbin/httpd -DFOREGROUND
           └─3582 /usr/sbin/httpd -DFOREGROUND

Oct 05 11:45:48 centos8.ittraining.loc systemd[1]: Starting The Apache HTTP Server...
Oct 05 11:45:48 centos8.ittraining.loc systemd[1]: Started The Apache HTTP Server.
Oct 05 11:45:48 centos8.ittraining.loc httpd[3578]: Server configured, listening on: port 443, port 80
```

Éditez maintenant le Connector HTTPS du fichier **$CATALINA_HOME/conf/server.xml** :

```
...
    <!-- <Connector port="8443" protocol="org.apache.coyote.http11.Http11NioProtocol"
               maxThreads="150" SSLEnabled="true">
        <UpgradeProtocol className="org.apache.coyote.http2.Http2Protocol" />
        <SSLHostConfig>
            <Certificate
            certificateKeystoreFile="/usr/tomcat10/security/www_ittraining_loc.keystore"
            certificateKeystorePassword="fenestros"
            type="RSA"
            />
        </SSLHostConfig>
    </Connector> -->

    <Connector port="8443" proxyPort="443" proxyName="10.0.2.45" />
...
```

Sauvegardez et relancez le service tomcat :

```
[root@centos8 ~]# systemctl restart tomcat
[root@centos8 ~]# systemctl status tomcat
● tomcat.service - Apache Tomcat Web Application Container
   Loaded: loaded (/etc/systemd/system/tomcat.service; disabled; vendor preset: disabled)
```

```
   Active: active (running) since Thu 2023-10-05 11:49:08 EDT; 8s ago
  Process: 3825 ExecStart=/usr/tomcat10/bin/startup.sh (code=exited, status=0/SUCCESS)
 Main PID: 3835 (java)
    Tasks: 63 (limit: 100949)
   Memory: 439.4M
   CGroup: /system.slice/tomcat.service
           └─3835 /usr/lib/jvm/jre-1.8.0-openjdk-1.8.0.312.b07-2.el8_5.x86_64/bin/java -
Djava.util.logging.config.file=/usr/tomcat10//conf/logging.properties -
Djava.util.logging.manager=org.apache.juli.ClassLo>

Oct 05 11:49:08 centos8.ittraining.loc systemd[1]: Starting Apache Tomcat Web Application Container...
Oct 05 11:49:08 centos8.ittraining.loc startup.sh[3825]: Existing PID file found during start.
Oct 05 11:49:08 centos8.ittraining.loc startup.sh[3825]: Removing/clearing stale PID file.
Oct 05 11:49:08 centos8.ittraining.loc startup.sh[3825]: Tomcat started.
Oct 05 11:49:08 centos8.ittraining.loc systemd[1]: Started Apache Tomcat Web Application Container.
```

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 :

```
[root@centos8 ~]# lynx --dump https://www.ittraining.loc/docs
   [1] Tomcat Home
   [2]The Apache Software Foundation

Apache Tomcat 10

   Version 10.0.27, Oct 3 2022

Links

     * [3]Docs Home
     * [4]FAQ
     * [5]User Comments
```

```
...
```

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@centos8 ~]# 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";

        permission java.util.PropertyPermission "sun.security.pkcs11.disableKeyExtraction", "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@centos8 ~]# 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;
};

// This permission is required when using javac to compile JSPs on Java 9
// onwards
//grant codeBase "jrt:/jdk.compiler" {
//        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, delete";

        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.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.*";

    // 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. It also requires the custom Tomcat
// DeployXmlPermission to enable the use of META-INF/context.xml
// 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:${catalina.base}/webapps/manager/-" {
    permission java.lang.RuntimePermission "accessClassInPackage.org.apache.catalina";
    permission java.lang.RuntimePermission "accessClassInPackage.org.apache.catalina.ha.session";
    permission java.lang.RuntimePermission "accessClassInPackage.org.apache.catalina.manager";
    permission java.lang.RuntimePermission "accessClassInPackage.org.apache.catalina.manager.util";
    permission java.lang.RuntimePermission "accessClassInPackage.org.apache.catalina.util";
    permission org.apache.catalina.security.DeployXmlPermission "manager";
};
grant codeBase "file:${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";
    permission org.apache.catalina.security.DeployXmlPermission "manager";
};

// The Host Manager application needs the custom Tomcat DeployXmlPermission to
// enable the use of META-INF/context.xml
// These settings support the following configurations:
// - default CATALINA_HOME == CATALINA_BASE
// - CATALINA_HOME != CATALINA_BASE, per instance Host Manager in CATALINA_BASE
// - CATALINA_HOME != CATALINA_BASE, shared Host Manager in CATALINA_HOME
grant codeBase "file:${catalina.base}/webapps/host-manager/-" {
    permission org.apache.catalina.security.DeployXmlPermission "host-manager";
};
grant codeBase "file:${catalina.home}/webapps/host-manager/-" {
    permission org.apache.catalina.security.DeployXmlPermission "host-manager";
};



// 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";
// };

// To grant permissions for web applications using packed WAR files, use the
// Tomcat specific WAR url scheme.
//
// The permissions granted to the entire web application
// grant codeBase "war:file:${catalina.base}/webapps/examples.war*/-" {
// };
//
// The permissions granted to a specific JAR
// grant codeBase "war:file:${catalina.base}/webapps/examples.war*/WEB-INF/lib/foo.jar" {
// };
```
