旧: インストール（IdPv4）

IdPv4のインストール

実習セミナー内に準備されたLinuxサーバにJDK、Tomcat、Shibboleth IdPをインストールする手順となっています。

---

1. 実習に使用する仮想サーバについて

以下は本技術ガイドで構築する前提となる環境です。

- OS、DNS、ネットワーク、時刻同期などは設定済みとなっています。（Apache HTTP Server、mod_sslもインストール済み）
- CentOS7
- メモリ2GB以上
- Apache HTTP Server 2.4 と mod_ssl
- Java 11（OpenJDK）
- Apache Tomcat 9
- Shibboleth IdP v4

また、実習環境ではSELinuxは無効化されているものとして手順を記載しています。下記コマンドでSELinux設定が確認できます。

```
$ /usr/sbin/getenforce
```

---

2. Java 11（OpenJDK）をインストールする

インストール

CentOS 7にはOpenJDKのパッケージが用意されていますので、これをyumにてインストールします。

```
# yum install java-11-openjdk java-11-openjdk-devel
```

---

3. Apache Tomcat 9をインストールする

1. インストール

CentOS 7に用意されているパッケージはTomcat7なので、Apache Software Foundationが配布するTomcatパッケージをダウンロードしてインストールします。
実習セミナーでは予めダウンロードした「/root/PKG」内の、apache-tomcat-9.?.??.tar.gzを使います。

```
# cd /root/PKG

# tar zxv -C /usr/share -f apache-tomcat-9.?.??.tar.gz
# ln -s /usr/share/apache-tomcat-9.?.?? /usr/share/tomcat
```

また自動起動スクリプトは、「/root/GETFILE」配下のtomcat.serviceを使います。

```
# cp /root/GETFILE/tomcat.service /etc/systemd/system
```

2. 自動起動の設定

以下のコマンドで自動起動設定を有効にします。

```
# systemctl enable tomcat


補足：

以下のコマンドで自動起動設定を無効にすることができます。

# systemctl disable tomcat
```

”tomcat”ユーザで起動

"root"ユーザではなく、Tomcat起動用のユーザを使用することを推奨します。
ここでは、一般的な"tomcat"ユーザを作成します。（以降、"tomcat"ユーザを使用する事が前提で説明します。）

```
# useradd -r -d /usr/share/tomcat -s /sbin/nologin -c ”Tomcat daemon” tomcat
```

以下のコマンドでその他Tomcat関連の設定ファイルやディレクトリの所有者、パーミッションを設定します。

```
# chown -R tomcat:tomcat /usr/share/tomcat/{temp,logs,work}

# chown tomcat:tomcat /usr/share/tomcat/webapps
# chmod +t /usr/share/tomcat/webapps
# chmod go+rx /usr/share/tomcat/conf
# chgrp tomcat /usr/share/tomcat/conf/*.*
# chmod g+r /usr/share/tomcat/conf/*.*
# mkdir -p /usr/share/tomcat/conf/Catalina/localhost
# chgrp -R tomcat /usr/share/tomcat/conf/Catalina
# chmod -R g+r /usr/share/tomcat/conf/Catalina
# chmod -R +t /usr/share/tomcat/conf/Catalina
# chgrp -R tomcat /usr/share/tomcat/{bin,lib}
```

3. JAVA_OPTSの設定

以下のように/etc/sysconfig/tomcatを修正します。

```
#JAVA_OPTS=”-Xminf0.1 -Xmaxf0.3”
JAVA_OPTS=”-server -Xmx1500m -XX:MaxPermSize=256m -XX:+UseG1GC ”
```

4. profileの追加

/etc/profile.d/java-tomcat.sh という新規ファイルを以下の内容で作成します。

```
# /etc/profile.d/java-tomcat.sh - set Java and Tomcat stuff
JAVA_HOME=/usr/lib/jvm/java
#export MANPATH=$MANPATH:/usr/java/default/man
CATALINA_HOME=/usr/share/tomcat
CATALINA_BASE=$CATALINA_HOME
PATH=$JAVA_HOME/bin:$CATALINA_BASE/bin:$CATALINA_HOME/bin:$PATH
export PATH JAVA_HOME CATALINA_HOME CATALINA_BASE
```

追加した環境変数を読み込みます。

```
# source /etc/profile
```

5. Apache の設定

以下のように /etc/httpd/conf/httpd.conf を修正します。

```
（省略）
#ServerName ex-idp-test??.gakunin.nii.ac.jp:80　← ??は各自割り振られた番番号（0番なら「00」）
↑コメントアウト（#）を削除
（省略）
```

以下のように /etc/httpd/conf.d/ssl.conf を修正します。

```
（省略）
<VirtualHost _default_:443>
（省略）
#ServerName ex-idp-test??.gakunin.nii.ac.jp:443　← ??は各自割り振られた番号（0番なら「00」）
↑コメントアウト（#）を削除
ProxyPass /idp/ ajp://localhost:8009/idp/ ← 設定を追加
（省略）
```

6. Tomcat の設定

以下の内容で /etc/httpd/conf.d/virtualhost-localhost80.conf を作成します。

※これはShibboleth IdPが提供するreload-metadata.sh等のコマンドを使った操作を可能にするためのものです。

```
<VirtualHost localhost:80>
ProxyPass /idp/ ajp://localhost:8009/idp/
</VirtualHost>
```

以下のように /usr/share/tomcat/conf/server.xml を修正します。

※実習環境では、Connector port="8080" をコメントアウトします。

```
<!--
    <Connector port="8080" protocol="HTTP/1.1"
               connectionTimeout="20000"
               redirectPort="8443" />
-->
```

※Connector port="8009"に以下のように追加してください。

```
<!-- Define an AJP 1.3 Connector on port 8009 -->
<!--
<Connector protocol="AJP/1.3"
address="::1"
port="8009"
redirectPort="8443" />
-->

<Connector port="8009" protocol="AJP/1.3" redirectPort="8443"
       secretRequired="false" enableLookups="false" tomcatAuthentication="false" address="127.0.0.1" maxPostSize="100000" />
```

---

4. Shibbolethのインストール

各ファイル名等の指定は，Version 4.0.1に準拠しています。

1. インストール

Shibboleth IdPのパッケージは、「/root/PKG」配下にあります。

以下のコマンドで移動してください。

```
# cd /root/PKG
```

shibboleth-identity-provider-4.?.?.tar.gz がすでに配置されているので、以下のコマンドを実行してください。

```
# tar xzvf shibboleth-identity-provider-4.?.?.tar.gz
# cd shibboleth-identity-provider-4.?.?
# ./bin/install.sh -Didp.conf.filemode=640
```

install.shシェルスクリプトを実行すると、以下のような問い合わせがあります。
手順に従って、進めてください。

インストール時に入力するパスワードを本運用で使う場合は、推測されにくいものを使用してください。
※ここで入力したパスワードは、/opt/shibboleth-idp/conf/idp.propertiesに記載されます。（平文）

```
Buildfile: /root/PKG/shibboleth-identity-provider-4.0.1/bin/build.xml

install:
Source (Distribution) Directory (press <enter> to accept default): [/root/PKG/shibboleth-identity-provider-4.0.1] ?
[Enter]  ←入力なし

Installation Directory: [/opt/shibboleth-idp]
[Enter]  ←入力なし
Hostname: [ex-idp-test??.gakunin.nii.ac.jp]
[Enter]  ←入力なし ※表示されたホスト名が違う場合、設定してください。
Backchannel PKCS12 Password: backpass[Enter]  ←任意のパスワード

Re-enter password: backpass[Enter]

Cookie Encryption Key Password: cookiepass[Enter]  ←任意のパスワード
Re-enter password: cookiepass[Enter]
SAML EntityID: [https://ex-idp-test??.gakunin.nii.ac.jp/idp/shibboleth]
[Enter]  ←入力なし
Attribute Scope: [gakunin.nii.ac.jp]
nii.ac.jp  [Enter]  ←nii.ac.jpを設定してください。

  （省略）

BUILD SUCCESSFUL
Total time: 2 minutes 9 seconds
```

上記のような質問に答えながら、インストールを行います。

2. パーミッションの調整

Tomcatを”tomcat”ユーザで実行する場合は、ログファイルを出力できるようディレクトリの所有者を変更します。
同様に、設定ファイルやメタデータの保存ディレクトリなどの所有者・パーミッションも変更します。

ここで設定したパーミッションをShibboleth IdPアップデート時に変更されないよう注意が必要です。詳細は IdPv3アップデートに関する情報 をご参照ください。

```
# chown -R tomcat:tomcat /opt/shibboleth-idp/logs
# chgrp -R tomcat /opt/shibboleth-idp/conf

# chmod -R g+r /opt/shibboleth-idp/conf

# find /opt/shibboleth-idp/conf -type d -exec chmod -R g+s {} ¥;
# chgrp tomcat /opt/shibboleth-idp/metadata
# chmod g+w /opt/shibboleth-idp/metadata
# chmod +t /opt/shibboleth-idp/metadata
# chgrp tomcat /opt/shibboleth-idp/credentials/secrets.properties

# chmod g+r /opt/shibboleth-idp/credentials/secrets.properties

# chgrp tomcat /opt/shibboleth-idp/credentials/sealer.*

# chmod g+r /opt/shibboleth-idp/credentials/sealer.*
```

IdPが実際に使用する証明書の秘密鍵はまだ配置されておりませんので、所有者・パーミッションは後の手順で設定します。

3. jstl-1.2.jar の配置

jstl-1.2.jarは、予め「/root/PKG」配下にあります。
edit-webapp/ 配下に配置し、idp.warに含めます。

```
# cp /root/PKG/jstl-1.2.jar /opt/shibboleth-idp/edit-webapp/WEB-INF/lib/
# /opt/shibboleth-idp/bin/build.sh
Installation Directory: [/opt/shibboleth-idp]
[Enter] ←入力なし
Rebuilding /opt/shibboleth-idp/war/idp.war ...
...done

BUILD SUCCESSFUL
Total time: 3 seconds
```

4. idp.war の登録

${CATALINA_BASE}/conf/Catalina/localhost/idp.xml という新規ファイルを以下の内容で作成し、idp.warをTomcatが認識できるようにします。

```
<Context docBase="/opt/shibboleth-idp/war/idp.war"
         privileged="true"
         antiResourceLocking="false"
         swallowOutput="true">

    <Manager pathname="" />

    <!-- Work around lack of Max-Age support in IE/Edge -->
    <CookieProcessor alwaysAddExpires="true" />

</Context>
```

httpdの再起動とTomcatの起動を行います。（すでにTomcatが起動している場合はstopしてから行ってください）

```
# systemctl restart httpd
# systemctl start tomcat
```

Tomcatの起動後、${CATALINA_BASE}/logs/catalina.{日付}.logにエラーが出力されていない事を確認してください。

ヒント

※catalina.{日付}.logにTomcat終了時（再起動時）のタイミングで以下のようなエラーが表示されることがありますが問題ありませんので無視してください。

```
致命的: A web application appears to have started a TimerThread named [Timer-0] via the java.util.Timer API but has failed to stop it. To prevent a memory leak, the timer (and hence the associated thread) has been forcibly cancelled.
```

```
致命的: A web application created a ThreadLocal with key of type [null] (value [ch.qos.logback.core.UnsynchronizedAppenderBase$1@XXXXXXXX]) and a value of type [java.lang.Boolean] (value [false]) but failed to remove it when the web application was stopped. To prevent a memory leak, the ThreadLocal has been forcibly removed.
```

(関連するバグレポート)

5. サービスの起動・停止方法

サービス	起動コマンド	停止コマンド	再起動コマンド
httpd	systemctl start httpd	systemctl stop httpd	systemctl restart httpd
tomcat	systemctl start tomcat	systemctl stop tomcat	systemctl restart tomcat

---

インストールが完了したら，サイト情報等の設定を行って下さい．
