Installazione Shibboleth IDP per Linux

10 Ottobre 2011

Autori: Marco Malavolti , Barbara Monticini e Francesco Malvezzi

Credits: Switch AAI, Shibboleth

Indice generale

1 Introduzione

Questo documento ha lo scopo di guidare l'utente nell'installazione di un IdP Shibboleth 2.3.3 con il solo Apache Tomcat 6.

2 Approfondimenti

- Installazione di Apache Tomcat da tar.gz
- Installazione di Java JDK con file autoestraente

3 Software da installare

- openssl;
- ntp;
- sun-java6-jdk (non-free);
- sun-java6-bin (non-free);
- tomcat6;
- expat (per il parsing di xml);
- tomcat6-dta-ssl-1.0.0.jar shib
- idp (http://www.shibboleth.net/downloads/identity-provider/latest/)

4 Richiedere il certificato per l'IDP

a) In linea con le **specifiche tecniche** della Federazione IDEM è necessario installare sulla porta 443 un certificato rilasciato da una CA riconosciuta. All'interno della comunità GARR è attivo il servizio di rilascio certificati server denominato **TCS** (TERENA Certificate Service). La caratteristica dei certificati TCS è quella di essere emessi da una CA commerciale che nello specifico consiste in **COMODO CA**.

   - L'elenco delle organizzazioni presso le quali il servizio TCS è gia' attivo è disponibile in https://ca.garr.it/TCS/tab.php
   - Se il servizio non fosse ancora attivo presso la vostra organizzazione è possibile contattare **GARR Certification Service** per avviare il procedimento di attivazione (e-mail a garr-ca@garr.it)

b) Per generare una richiesta di certificato seguire le istruzioni suggerite nelle pagine di documentazione TCS (https://ca.garr.it/TCS/doc_server.php)

c) Le richieste di certificato devono essere inviate ai **referenti TCS** presenti nella vostra organizzazione (denominati Contatti Amministrativi TCS). Per conoscere i nomi dei Contatti Amministrativi nominati all'interno del vostro Ente inviare una mail di richiesta a garr-ca@garr.it

5 Installare Tomcat 6, sun-java6-jdk, sun-java6-bin, openssl e ntp[1]

a) `sudo apt-get install sun-java6-jdk`[2] `sun-java6-bin ca-certificates openssl ntp tomcat6`

b) `sudo nano /etc/environment` e aggiungere queste righe in fondo al file:

```
TOMCAT_HOME=/usr/share/tomcat6

CATALINA_OUT=/var/log/tomcat6/catalina.out
CATALINA_HOME=/var/lib/tomcat6

JAVA_HOME=/usr/lib/jvm/java-6-sun-1.6.0.26

JAVA_OPTS="-Djava.awt.headless=true -Xmx512M -XX:MaxPermSize=128m"
```

(In questo modo si configura la memoria della JVM per esaudire l'IdP Web Application. Il valore per la memoria usata dipende dalla memoria fisica del server. Impostare Xmx (massimo heap space a disposizione della JVM) ad almeno 512MB e XX:MaxPermSize a 128 MB.)

d) Fare Logout e Login per attuare i cambiamenti all'environment della macchina

e) `/etc/init.d/tomcat6 start`

Passo Facoltativo) Amministrare Tomcat da http://localhost:8080/manager/html:

a) `sudo su -`
b) `apt-get install tomcat6-admin`
b) `nano $CATALINA_BASE/conf/tomcat-users.xml`
c) aggiungere le seguenti righe tra i tag `<tomcat-users> ... </tomcat-users>`:

```
<tomcat-users>
      ...
      <role rolename="manager"/>
      <role rolename="administrator"/>

      <user username="Admin" password="password_Amministratore"
            roles="admin,manager"/>

      <user username="Manager" password="password_Manager"
            roles="manager"/>

</tomcat-users>
```

[1] per ubuntu 10.04 e superiori
[2] se il sistema monta la openJDK è necessario rimuoverla completamente, comprese le sue dipendenze

6 Installare Shibboleth Identity Provider Shibboleth 2.3.3

a) `sudo su -`

b) `cd /usr/local/src`

c) `wget http://www.shibboleth.net/downloads/identity-provider/2.3.3/shibboleth-identityprovider-2.3.3-bin.zip`

d) `unzip shibboleth-identityprovider-2.3.3-bin.zip`

e) `cd shibboleth-identityprovider-2.3.3/`

f) `sh install.sh` [annotarsi e scegliere l'Identity Provider Server (Predefinito: "`idp.example.org`") e la sua password (Es. 123456)]
(Lasciare come percorso di installazione quello predefinito "`/opt/shibboleth-idp`")

g) Copiare le librerie di Xerces (Java parser for XML) e di Xalan (Xalan è un XSLT processor per trasformare documenti XML in documenti HTML,testo, o altri documenti XML) da Shibboleth IdP source package in `$TOMCAT_HOME/endorsed`.

Copiare la cartella "`endorsed/`" nella cartella `$TOMCAT_HOME` così:

```
cp -r /usr/local/src/shibboleth-identityprovider-2.3.3/endorsed/
$TOMCAT_HOME
```

h) `sudo gedit /etc/environment` e aggiungere queste righe in fondo al file:

```
JAVA_ENDORSED_DIRS=/usr/share/tomcat6/endorsed

IDP_HOME=/opt/shibboleth-idp
```

i) Fare Logout e Login per attuare i cambiamenti all'environment della macchina

j) Aggiustare i permessi dell'utente tomcat6 sulla directory dell'idp:

```
chown tomcat6:nogroup /opt/shibboleth-idp/logs/
chown tomcat6:nogroup /opt/shibboleth-idp/metadata/
chown tomcat6:nogroup /opt/shibboleth-idp/credentials/
```

7 Configurare Tomcat

a) `sudo su -`

b) `cd $TOMCAT_HOME/lib`

c) `wget http://shibboleth.internet2.edu/downloads/maven2/edu/internet2/middleware/security/tomcat6/tomcat6-dta-ssl/1.0.0/tomcat6-dta-ssl-1.0.0.jar`

d) `nano $CATALINA_HOME/conf/server.xml` e aggiungere il seguente connettore:

```
<Connector port="8443"
           protocol="org.apache.coyote.http11.Http11Protocol"
           SSLImplementation="edu.internet2.middleware.security.
               tomcat6.DelegateToApplicationJSSEImplementation"
           scheme="https"
           SSLEnabled="true"
           clientAuth="true"
           keystoreFile="/opt/shibboleth-idp/credentials/idp.jks"
           keystorePass="yourPasswordHere" />
```

(al posto di `yourPasswordHere` mettere la password scelta durante l'installazione dell'IdP, questo connettore è utile ai soli fini di retrocompatibilità con SAML 1.x)

e) Decommentare il Connector alla porta 8443 già presente e cambiare la sua porta da `<Connector port=8443 ...` in `<Connector port=443 …`

f) Cambiare la porta del `<Connector port=8080 ...` in `<Connector port=80 …`

g) `nano /etc/default/tomcat6` e, in fondo al file, decommentate la riga "`#AUTHBIND=no`" e modificatela in "`AUTHBIND=yes`" in questo modo TOMCAT potrà usare le porte superiori a 1024.

h) Completare il certificato richiesto per il Server con la catena CA prelevabile da qui: https://ca.garr.it/mgt/Terena-chain.pem

(per "completare" si intende copiare e incollare il contenuto del file Terena-chain.pem nel certificato del server rilasciato dalla CA sotto la scritta "-----END CERTIFICATE-----")

i) Posizionare la chiave privata utilizzata per la creazione del Certificato del Server (**key-server.pem**) e il certificato del server che è stato rilasciato (cert.pem - es.: **cert-9999-prova.lab.test.it.pem**) in una cartella a parte (es.: .../Certificati), posizionarsi su di essa e digitare il seguente comando:

```
openssl pkcs12 -export -in cert.pem -inkey key-server.pem -out server-full-cert.p12
```

j) `keytool -importkeystore -destalias idp2-tomcat-ssl -destkeystore $IDP_HOME/credentials/idp2.ssl.jks -srckeystore server-full-cert.p12 -srcstoretype PKCS12 -alias 1`

(Digitare le password che vi vengono richieste)

k) `keytool -list -keystore idp2.ssl.jks`
(e memorizzate su un foglio di carta il **keyAlias**, il **keystoreFile** e il **keyPass**)

Otterrete qualcosa di simile a questo:

```
root# keytool -list -keystore idp2.ssl.jks
Immettere la password del keystore:


Tipo keystore: JKS
Provider keystore: SUN


Il keystore contiene 1 entry


idp2-tomcat-ssl, 26-set-2011, PrivateKeyEntry,
Impronta digitale certificato (MD5):
23:xx:5E:yy:8B:zz:4D:hh:71:bb:F8:gg:D5:ll:8B:rr
```

(xx, yy, zz, hh, bb, gg, ll e rr sono Valori nascosti per sicurezza)

dove `idp2-tomcat-ssl` sarà il **keyAlias** (**keyAlias**="`idp2-tomcat-ssl`")

in `$IDP_HOME/credentials/` troverete il file "`idp2.ssl.jks`", il **keystoreFile**

il **keypass** sarà la password scelta per l'IdP durante la sua installazione (**keypass**="`yourPasswordHere`")

l) Ora che avete trovato queste 3 informazioni importanti (**keyAlias**, **keystoreFile** e **keypass**) dovrete aggiungerle al `<Connector port=443 ...` in modo che risulti simile a questo frammento di codice nel file `$TOMCAT_HOME/conf/server.xml`:

```
<Connector port="443"
           protocol="HTTP/1.1"
           SSLEnabled="true"
           maxThreads="150"
           scheme="https" secure="true"
           clientAuth="false" sslProtocol="TLS"
           keyAlias="idp2-tomcat-ssl"
           keystoreFile="/opt/shibboleth-idp/credentials/idp2.ssl.jks"
           keypass="yourPasswordHere"/>
```

(modificare adeguatamente i valori di **keyAlias**, **keystoreFile** e **keypass** prima di proseguire!!!)

m) Potete verificare che i certificati siano installati correttamente digitando:
`openssl s_client -CApath /etc/ssl/certs/ -connect idp.lab.test.it:443`

e ricevendo "`Verify return code: 0 (ok)`"

n) Depositare l' IdP WAR file, localizzato in `$IDP_HOME/war/` usando un context deployment fragment:

La normale procedura per il deploying delle Web Application in Tomcat è attuata mediante la copia del file WAR nella cartella `webapps/` di Tomcat.

Tuttavia, quando questa procedura viene eseguita, Tomcat espande il WAR file (ottenendo così il file `idp/` nella cartella `webapps/` ma senza cancellare il file WAR) e carica la nuova versione dell'applicazione in "`work/Catalina/localhost/`". Questo può causare l'utilizzo di una precedente versione del WAR anche se viene copiata una versione nuova nella giusta posizione (`webapps/`).

Per ovviare a questo inconveniente, viene raccomandato di usare un context deployment fragment. Questo significa che si userà un piccolo pezzo di XML per dire a Tomcat dove andare a prendere il WAR e fornire qualche proprietà da usare quando Tomcat caricherà l'applicazione.

o) `sudo nano /etc/tomcat6/Catalina/localhost/idp.xml` e copiarvi dentro questo pezzo di codice:

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

p) Salvare e riavviare tomcat (`sudo /etc/init.d/tomcat restart`)

q) Installazione dell'IdP conclusa, testiamolo!

Aggiungere al file `/etc/hosts` la seguente riga:

```
127.0.0.2   idp.lab.test.it
```

Aprite 1 finestra del Browser e digitate:
`https://idp.lab.test.it:443/idp/profile/Status`
e deve darvi OK. ==> IdP funzionante su HTTPS

Aprite 1 finestra del Browser e digitate:
`https://idp.lab.test.it:8443/idp/profile/Status`
e deve darvi ERRORE ==> IdP che richiede il certificato

N.B.: Ogni volta che si cambia WAR in `/opt/shibboleth-idp/war` facendo il suo Undeploy da Tomcat Manager o altro, BISOGNA ricordarsi di ricreare l'**idp.xml** dentro a `/etc/tomcat6/Catalina/localhost/` che indica a Tomcat di prendere il nuovo WAR.

8 Installazione di Shibboleth Security Provider

a) `sudo su -`

b) `cd shibboleth-identityprovider-2.3.3/lib/`

c) `cp -v shibboleth-jce-1.1.0.jar $JAVA_HOME/jre/lib/ext/`

d) `nano /etc/java-6-sun/security/java.security` e aggiungere questa riga all'elenco dei suoi simili:

```
security.provider.#=edu.internet2.middleware.shibboleth.Delegate-
ToApplicationProvider
```

(dove # deve essere sostituto con il **primo** numero progressivo disponibile - `security.provider.9=edu.internet2.middleware.shibboleth.DelegateToApplica-tionProvider`)

9 Specificare per Tomcat l'equivalente di OPTIONAL_NO_CA di Apache2

a) `sudo su -`

b) `cd shibboleth-identityprovider-2.3.3/lib/`

c) `cp -v shibboleth-jce-1.1.0.jar $JAVA_HOME/jre/lib/ext/`

d) `nano /etc/java-6-sun/security/java.security` e aggiungere questa riga all'elenco dei suoi simili:

```
security.provider.#=edu.internet2.middleware.shibboleth.quickInsta
llIdp.AnyCertProvider
```

(dove # deve essere sostituto con il **primo** numero progressivo disponibile -

`security.provider.10=edu.internet2.middleware.shibboleth.quickInstallIdp.A nyCertProvider)`

10 Configurare Shibboleth Identity Provider

a) `sudo su -`

b) `nano /etc/environment` e aggiungere le seguenti variabili che torneranno utili:

```
IDP_LOG=/opt/shibboleth-idp/logs/idp-process.log

TOMCAT_LOG_DIR=/var/log/tomcat6/
```

c) `nano /opt/shibboleth/conf/logging.xml` e modificarlo in modo da ottenere la seguente parte iniziale:

```
<!-- Logs IdP, but not OpenSAML, messages -->
<logger name="edu.internet2.middleware.shibboleth" level="DEBUG"/>

<!-- Logs OpenSAML, but not IdP, messages -->
<logger name="org.opensaml" level="DEBUG"/>

<!-- Logs LDAP related messages -->
<logger name="edu.vt.middleware.ldap" level="DEBUG"/>

<!-- Logs inbound and outbound protocols messages at DEBUG level-->
<logger name="PROTOCOL_MESSAGE" level="DEBUG" />
```

d) `apt-get install expat`
(servirà per verificare la correttezza degli XML con il comando `xmlwf nomeFile.xml`)

e) `nano $IDP_HOME/conf/handler.xml` e

1. Commentare il blocco relativo all'endpoint `RemoteUser`
2. Abilitare il blocco relativo all'endpoint `UsernamePassword` (decommendandolo)

f) Modificare il file di configurazione `login.config` (`/opt/shibboleth-idp/conf/login.config`) come segue:

1. **Esempio** di connessione a LDAP senza SSL:

```
edu.vt.middleware.ldap.jaas.LdapLoginModule required
host="ldap://gt3.fi.infn.it:389"
base="dc=garr,dc=it"
serviceCredential="password_serviceUser"
serviceUser="cn=ldapadmin,dc=garr,dc=it"
ssl="false"
userField="uid"
subtreeSearch="true";
```

2. **Esempio** di connessione LDAP con SSL:

```
edu.vt.middleware.ldap.jaas.LdapLoginModule required
host="ldap://gt3.fi.infn.it:389"
base="dc=garr,dc=it"
serviceCredential="password_serviceUser"
serviceUser="cn=ldapadmin,dc=garr,dc=it"
ssl="true"
userField="uid"
subtreeSearch="true";
```

3. **Esempio** di connessione LDAP con TLS:

```
edu.vt.middleware.ldap.jaas.LdapLoginModule required
host="ldap://gt3.fi.infn.it:389"
base="dc=garr,dc=it"
serviceCredential="password_serviceUser"     serviceUser="cn=lda-
padmin,dc=garr,dc=it"
tls="true"
userField="uid"
subtreeSearch="true";
```

g) Gestire `eduPersonTargetID` come tipo `StoredID`:

1. `apt-get istall mysql-admin mysql-server libmysql-java`
2. `ln -s /usr/share/java/mysql-connector-java.jar /var/lib/tomcat6/common/`
3. `ln -s /usr/share/java/mysql-connector-java.jar /opt/shibboleth-idp/lib/`
4. `mysql -u root -p` /* necessario per accedere come root a mysql */
5. `SELECT User, Host, Password FROM mysql.user;`
   /* Tabella degli Utenti permessi */
6. `SET PASSWORD FOR 'root'@'localhost' = PASSWORD('newpwd');`
   /* Impostare 1 password di root per ogni riga in cui compare nel comando precedente */
7. in `mysql>` digitare " `create database userdb;` " per creare il database userdb di test. Vi restituirà: " `Query OK, 1 row affected (0.00 sec)` "
8. in `mysql>` digitare " `grant all privileges on userdb.* to root@localhost identified by 'yourPassword';` "

9. `use userdb;` /* Così gli dico di usare il database che abbiamo creato prima */

10.
```
mysql> CREATE TABLE shibpid
    -> (
    -> localEntity VARCHAR(255) NOT NULL,
    -> peerEntity VARCHAR(255) NOT NULL,
    -> principalName VARCHAR(255) NOT NULL,
    -> localId VARCHAR(255) NOT NULL,
    -> persistentId VARCHAR(255) NOT NULL,
    -> peerProvidedId VARCHAR(255) NULL,
    -> creationDate TIMESTAMP NOT NULL,
    -> deactivationDate TIMESTAMP NULL,
    -> KEY persistentId (persistentId),
    -> KEY persistentId_2 (persistentId, deactivationDate),

    -> KEY localEntity (localEntity(16), peerEntity(16),localId),
    -> KEY localEntity_2 (localEntity(16), peerEntity(16), localId,
    -> deactivationDate)
    -> );
```

11. `nano $IDP_HOME/conf/attribute-resolver.xml` e decommentare tutti gli `<resolver:AttributeDefinition .... >` fino a:

```
<resolver:AttributeDefinition
         xsi:type="ad:SAML2NameID"
         id="eduPersonTargetedID"
         nameIdFormat="urn:oasis:names:tc:SAML:2.0:nameid-
                       format:persistent"
         sourceAttributeID="computedID">

    <resolver:Dependency ref="computedID" />
    <resolver:AttributeEncoder
              xsi:type="enc:SAML1XMLObject"
              name="urn:oid:1.3.6.1.4.1.5923.1.1.1.10" />

    <resolver:AttributeEncoder
              xsi:type="enc:SAML2XMLObject"
              name="urn:oid:1.3.6.1.4.1.5923.1.1.1.10"
              friendlyName="eduPersonTargetedID" />

</resolver:AttributeDefinition>
```

E modificare tutte le stringhe:
"**computedID**" → "**storedID**"

12. NON DECOMMENTARE IL SEGUENTE FRAMMENTO DI CODICE:

```
<!-- Do NOT use the version of eduPersonTargetedID defined below
unless you understand why it was deprecated and know that this
reason does not apply to you.




<!--

<resolver:AttributeDefinition xsi:type="ad:Scoped" idID.old" sco-
pe="unimo.it" sourceAttributeID="persistentID">

<resolver:Dependency ref="storedID" />

<resolver:AttributeEncoder xsi:type="enc:SAML1ScopedString"
name="urn:mace:dir:attribute-def:eduPersonTargetedID" />

</resolver:AttributeDefinition>

-->
```

13. Personalizzare i `<resolver:DataConnector ... />` in modo che:

```
<resolver:DataConnector id="myLDAP"
                        xsi:type="LDAPDirectory"
                        ldapURL="ldap://ramo1.di.ldap.da.reperire
                                 ldap://ramo2.di.ldap.da.reperire"
                        starttls="true"
                        baseDN="dc=dc_di_ldap,dc=dc_di_ldap"


                        /* Parametri omissibili */
                        principal="cn=admin_di_ldap,dc=garr,dc=it"
                        principalCredential="password_principal">
                        /************************/


                        <dc:FilterTemplate>
                              <![CDATA[
                              (uid=$requestContext.principalName)
                              ]]>
                        </dc:FilterTemplate>
                  </resolver:DataConnector>


<resolver:DataConnector xsi:type="StoredId"
                     xmlns="urn:mace:shibboleth:2.0:resolver:dc"
                     id="storedID"
                     sourceAttributeID="cn"
                     generatedAttributeID="persistentID"
```

```
                    salt="digitare_stringa_casuale_di_numeri_let
                          tere_simboli">

     <resolver:Dependency ref="myLDAP" />

     <ApplicationManagedConnection
                 jdbcDriver="com.mysql.jdbc.Drive”
                 jdbcURL="jdbc:mysql://localhost:3306/nome_databa
                          se_creato_-_userdb_in_questo_caso"
                 jdbcUserName="db_user_-_root_in_questo_caso"
                 jdbcPassword="password_db_user" />
</resolver:DataConnector>
```

14. Modificare il file `attribute-filter.xml` per permettere all'IdP di inviare qualche attributo, per esempio:

```
<!--  Release the transient ID to anyone -->
<afp:AttributeFilterPolicy id="releaseTransientIdToAnyone">
      <afp:PolicyRequirementRule xsi:type="basic:ANY"/>

      <afp:AttributeRule attributeID="transientId">
            <afp:PermitValueRule xsi:type="basic:ANY"/>
      </afp:AttributeRule>

      <afp:AttributeRule attributeID="eduPersonTargetedID">
            <afp:PermitValueRule xsi:type="basic:ANY" />
      </afp:AttributeRule>
</afp:AttributeFilterPolicy>

<afp:AttributeFilterPolicy id="attributesToAnyone">
      <afp:PolicyRequirementRule xsi:type="basic:ANY"/>

      <afp:AttributeRule attributeID="commonName">
            <afp:PermitValueRule xsi:type="basic:ANY"/>
      </afp:AttributeRule>

      <afp:AttributeRule attributeID="email">
            <afp:PermitValueRule xsi:type="basic:ANY"/>
      </afp:AttributeRule>

      <afp:AttributeRule attributeID="surname">
            <afp:PermitValueRule xsi:type="basic:ANY"/>
```

```
      </afp:AttributeRule>
</afp:AttributeFilterPolicy>
```

h) `sudo /etc/init.d/tomcat6 restart`

i) `nano $IDP_HOME/conf/relying_party.xml` e modificarlo **esattamente** come da email ricevuta dopo aver spedito il proprio `/opt/shibboleth-idp/metadata/idp-metadata.xml` a idem-help@garr.it

j) Verificare che compaia la pagina di Login dopo essere acceduti al proprio IDP dalla pagina del Service Provider di Test (https://sp-test.garr.it) inviata con la mail ricevuta dall' idem-help@garr.it

11 Approfondimenti

11.1 Installare la Sun Java JDK da pacchetto autoestraente (.bin):

a) Aprire il Gestore pacchetti e rimuovere openJDK
b) `sudo su`
c) `mkdir /home/nome_utente/Scrivania/sun-java_6u27`
d) `cd Scrivania/sun-java_6u27`
e) Download Sun Java JDK 6u27 – 32 o 64 bit:

   1. Versione per x86: `wget http://download.oracle.com/otn-pub/java/jdk/6u27-b07/jdk-6u27-linux-i586.bin`

   2. Versione per x64: `wget http://download.oracle.com/otn-pub/java/jdk/6u27-b07/jdk-6u27-linux-x64.bin`

f) Installazione Java 6u27:
   1. `chmod +x jdk-6u27-linux-i586.bin` o `chmod +x jdk-6u27-linux-x64.bin`
   2. `./jdk-6u27-linux-i586.bin` o `./jdk-6u27-linux-x64.bin`

g) `sudo mv /home/nome_utente/Scrivania/sun-java_6u27/jdk1.6.0_27 /usr/local/src`

h) `sudo gedit /etc/environment` e aggiungere queste righe in fondo al file:

```
JAVA_HOME="/usr/local/src/jdk1.6.0_27"

PATH="$PATH:/usr/local/src/jdk1.6.0_27/bin"
```

i) fare logout e login o riavviare per applicare le modifiche

j) `java -version` (per controllare la correttezza dell'operato)

11.2 Installazione Tomcat v6.0.33 da sorgente:

a) Installare la Sun Java JDK 1.6.0_6u27

b) `cd Scrivania`

c) `wget http://apache.panu.it/tomcat/tomcat-6/v6.0.33/bin/apache-tomcat-6.0.33.tar.gz`

d) `tar -xf Scaricati/apache-tomcat-6.0.33.tar.gz`

e) `sudo mv apache-tomcat-6.0.33/ /usr/local/src`

f) `sudo ln -s /usr/local/src/apache-tomcat-6.0.33/ /usr/local/tomcat`

g) `sudo nano /etc/environment` e aggiungere queste righe in fondo al file:

```
CATALINA_HOME=/usr/local/tomcat

CATALINA_BASE=/usr/local/tomcat
```

h) `sudo gedit $CATALINA_HOME/bin/catalina.sh` e aggiungere la seguente riga:

```
JAVA_OPTS="-Djava.awt.headless=true -Xmx512M -XX:MaxPermSize=128M
-Dcom.sun.security.enableCRLDP=true"
```

(In questo modo si configura la memoria della JVM per esaudire l'IdP Web Application. Il valore per la memoria usata dipende dalla memoria fisica del server.
Imposto l' Xmx (massimo heap space a disposizione della JVM) ad almeno 512MB e il XX:MaxPermSize a 128 MB.)

i) `sudo gedit $CATALINA_HOME/conf/server.xml` e modificare la seguente riga:

```
<Host appBase="webapps"
      unpackWARs="true"
      autoDeploy="true"
      xmlValidation="false"
      xmlNamespaceAware="false">
```

in:

```
<Host appBase="webapps"
      unpackWARs="true"
      autoDeploy="false"
      xmlValidation="false"
      xmlNamespaceAware="false">
```

j) Creare un file, sulla Scrivania, di nome "tomcat" con un editor di testo contenente questo:

```
#
# Startup script for the Tomcat server
#
# chkconfig: - 83 53
# description: Starts and stops the Tomcat daemon.
# processname: tomcat
# pidfile: /var/run/tomcat.pid
# See how we were called.
case $1 in
        start)
           export JAVA_HOME=/usr/local/src/jdk1.6.0_27
           export CLASSPATH=/usr/local/tomcat/lib/servlet-api.jar
           export CLASSPATH=/usr/local/tomcat/lib/jsp-api.jar
           export JRE_HOME=/usr/local/src/jdk1.6.0_27/jre
           echo "Tomcat is started"
           sh /usr/local/tomcat/bin/startup.sh
        ;;
        stop)
           export JRE_HOME=/usr/local/src/jdk1.6.0_27/jre
           sh /usr/local/tomcat/bin/shutdown.sh
           echo "Tomcat is stopped"
        ;;
        restart)
           export JRE_HOME=/usr/local/src/jdk1.6.0_27/jre
           sh /usr/local/tomcat/bin/shutdown.sh
           echo "Tomcat is stopped"
           sh /usr/local/tomcat/bin/startup.sh
           echo "Tomcat is started"
        ;;
        *)
           echo "Usage: /etc/init.d/tomcat start|stop|restart"
        ;;
esac
exit 0
```

k) `sudo mv Scrivania/tomcat /etc/init.d`
l) `sudo chmod 755 /etc/init.d/tomcat`
m) `sudo update-rc.d tomcat defaults`
n) `sudo /etc/init.d/tomcat start`

Passo Facoltativo) Amministrare Tomcat da Interfaccia Grafica:

a) `sudo su`

b) `nano $CATALINA_BASE/conf/tomcat-users.xml`

c) aggiungere le seguenti righe tra i tag `<tomcat-users> ... </tomcat-users>`:

```
<tomcat-users>
      ...
      <role rolename="manager"/>
      <role rolename="administrator"/>

      <user username="Admin" password="password_Amministratore"
            roles="admin,manager"/>

      <user username="Manager" password="password_Manager"
            roles="manager"/>
      </tomcat-users>
```

d) Salvare e riavviare Tomcat con "`sudo /etc/init.d/tomcat restart`"

e) Accedere alla pagina "`localhost:8080/manager/html`" con Username e Password definiti sopra.