## Posada en marxa

Per tenir l'aplicació en funcionament haurem de dur a terme dos processos: la posada en marxa manual mitjançant un *script* i l'execució de la funció d'inicialització dins de la secció d'administrador de l'aplicació.

La posada en marxa es farà a través d'un *script* (CAuoc.sh) i d'un fitxer de configuració (user.conf) que l'usuari podrà modificar per personalitzar els paràmetres del programa. Tant l'*script* com el fitxer de configuració els trobarem a la carpeta *bin* de l'aplicació.

Aquesta posada en marxa preveu tres passes corresponents a les opcions de l'*script*:

**sh CAuoc.sh -mysql**

Crea un usuari de mySQL per a l'aplicació. No cal crear la base de dades o les taules que farem servir, ja que se n'encarrega l'aplicació amb el procediment d'inicialització.

**sh CAuoc.sh -novaca**

Crea l'estructura de carpetes i el certificat de la CA, la seva clau privada i un *keystore* que conté el certificat.

**sh CAuoc.sh -tomcat**

Crea un certificat per a Tomcat i el signa per la CA creada anteriorment (Cal haver executat l'opció -novaca abans de fer servir aquesta). Aquest certificat, guardat en un magatzem de claus, es farà servir per poder accedir a l'aplicació mitjançant una connexió segura. També crearà un magatzem de confiança per a identificar els certificats de la CA que vulguin entrar a l'aplicació com a administrador.

El fitxer de configuració **user.conf** és un fitxer de text que conté les opcions que faran servir tant l'*script* com l'aplicació. Per tant és important no modificar aquest fitxer un cop s'ha inicialitzat el sistema, ja que les dades que rebi l'aplicació seran incorrectes i caldrà reinicialitzar-la.

El format del fitxer és del tipus [clau] = [valor]. Cal respectar aquest format i no canviar els noms de les claus per evitar errors durant l'execució. Tanmateix cal que els valors no incloguin espais ni símbols especials.

Els valors per defecte del fitxer són els següents. Entre parèntesi hi ha una explicació de cada clau:

```
capassword = 8g90A3kpv4
(Contrasenya del magatzem de claus i la clau privada de la CA)
```

tomcatpassword = 3rt78jiK49
(Contrasenya del magatzem de claus i la clau privada del certificat de Tomcat)

mysqluser = dbadmin
(Nom de l'usuari de mySQL per a l'aplicació)

mysqlpassword = dik827dg3K
(Contrasenya de l'usuari de mySQL per a l'aplicació)

localuser = pkiadmin
(Nom de l'usuari per a connexions locals a l'aplicació)

localpassword = uoc
(Contrasenya de l'usuari per a connexions locals a l'aplicació)

host = localhost:8443/tfcpki
(Host de l'aplicació. El host cal que inclogui el port i la carpeta de l'aplicació, però no l'inici d'adreça “https://”)

certdays = 365
(Número de dies de validesa dels nous certificats)

crldays = 30
(Número de dies de validesa de la CRL)

És necessari que existeixi el següent programari instal·lat al servidor per poder dur a terme la posada en marxa i l'execució. Entre parèntesi es dóna la versió feta servir pel desenvolupament:

- Apache Tomcat 7 (7.0.11)
- Java SE 1.6 (1.6.0.24_b07)
- OpenSSL (0.9.8k 25 Mar 2009)
- MySQL 5.1 (5.1.41-3ubuntu12.10)

En els següents apartats s'especifiquen les ordres per posar en marxa l'aplicació manualment.

Dins la resposta del *shell* està marcat en negreta on la intervenció de l'usuari és necessària.

## Creació de l'usuari de mySQL per a l'aplicació

Per que l'aplicació pugui fer servir una base de dades mySQL, haurem de crear un usuari perquè s'hi connecti.

Ho farem amb la següent opció de l'*script,* executant-lo des d'una consola:

***Comanda:***
sh CAuoc.sh -mysql

Durant l'execució se'ns demanarà la contrasenya de *root* per tenir prou privilegis per crear l'usuari.

## Creació dels certificats i claus de la CA

Per dur a terme la creació del certificat auto-signat i claus de la CA farem servir l'opció -novaca de l'*script* des d'una consola.

Durant l'execució es demanarà la intervenció de l'usuari per introduir les dades de la CA *(Distinguished Name).* Es poden acceptar les opcions per defecte entre corxets. També se'ns demanarà si volem fer el certificat confiable, opció que haurem de respondré afirmativament (*yes*).

***Comanda:***
sh CAuoc.sh -novaca

***Resposta:***
```
user@host:.../tfcpki/bin$ sh CAuoc.sh -novaca
Creant el certificat de la CA ...
Generating a 2048 bit RSA private key
.................+++
.....................................................................................
.............................+++
unable to write 'random state'
writing new private key to './ca/privat/./cakey.pem'
-----
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) [ES]:
State or Province Name (full name) [Barcelona]:
Locality Name (eg, city) [Barcelona]:
Organization Name (eg, company) [UOC]:
Organizational Unit Name (eg, section) []:ETIS
Common Name (eg, YOUR name) []:CA
Email Address []:lluismoran@uoc.edu
Owner: EMAILADDRESS=lluismoran@uoc.edu, CN=CA, OU=ETIS, O=UOC, L=Barcelona,
ST=Barcelona, C=ES
Issuer: EMAILADDRESS=lluismoran@uoc.edu, CN=CA, OU=ETIS, O=UOC, L=Barcelona,
ST=Barcelona, C=ES
Serial number: b5cf82286eff9ded
Valid from: Sun May 01 20:48:55 CEST 2011 until: Fri Apr 29 20:48:55 CEST
2016
Certificate fingerprints:
    MD5: 81:41:83:E6:D7:F7:A5:AD:7B:AE:8A:5C:4B:A0:4B:DA
```

```
    SHA1: 42:18:48:D7:08:01:43:38:69:26:84:22:42:70:13:55:7E:23:EC:DA
    Signature algorithm name: SHA1withRSA
    Version: 3

Extensions:

#1: ObjectId: 2.5.29.14 Criticality=false
SubjectKeyIdentifier [
KeyIdentifier [
0000: 95 35 DC 46 21 F0 39 E0  17 30 E8 4D 05 C7 FC 22 .5.F!.9..0.M..."
0010: 21 82 0F 42                                       !..B
]
]

#2: ObjectId: 2.5.29.19 Criticality=false
BasicConstraints:[
  CA:true
  PathLen:2147483647
]

#3: ObjectId: 2.5.29.35 Criticality=false
AuthorityKeyIdentifier [
KeyIdentifier [
0000: 95 35 DC 46 21 F0 39 E0   17 30 E8 4D 05 C7 FC 22  .5.F!.9..0.M..."
0010: 21 82 0F 42                                        !..B
]

[EMAILADDRESS=lluismoran@uoc.edu, CN=CA, OU=ETIS, O=UOC, L=Barcelona, ST=Bar-
celona, C=ES]
SerialNumber: [    b5cf8228 6eff9ded]
]

Trust this certificate? [no]:  yes
Certificate was added to keystore
```

## Creació dels certificat de Tomcat i signatura per part de la nostra CA

Per augmentar la seguretat, l'aplicació només accepta connexions segures (HTTPS). Per això necessitarem instal·lar al servidor Tomcat un certificat signat per la nostra CA.

Farem servir l'opció -tomcat de l'*script* des d'una consola.

Durant l'execució és demanarà la intervenció de l'usuari per confirmar algunes opcions. Caldrà picar *Enter* en la primera qüestió i respondre afirmativament a les altres quatre (*yes*)

***Comanda:***

sh CAuoc.sh -tomcat

***Resposta:***

```
user@host:.../tfcpki/bin$ sh CAuoc.sh -tomcat
Creant el certificat de Tomcat ...
Premeu Enter quan us demani si voleu la mateixa contrassenya...
Enter key password for <tomcat>
   (RETURN if same as keystore password):
```

```
Using configuration from ./ca.cnf
Check that the request matches the signature
Signature ok
Certificate Details:
        Serial Number: 0 (0x0)
        Validity
            Not Before: May  1 18:54:38 2011 GMT
            Not After : Apr 30 18:54:38 2012 GMT
        Subject:
            countryName               = ES
            stateOrProvinceName       = Barcelona
            organizationName          = UOC
            organizationalUnitName    = ETIS
            commonName                = Tomcat
        X509v3 extensions:
            X509v3 Basic Constraints:
                CA:FALSE
            X509v3 Subject Key Identifier:
                08:AD:6C:79:45:C7:22:A7:E7:0F:15:7F:B9:1E:78:24:EB:6E:F8:2A
            X509v3 Authority Key Identifier:
                keyid:95:35:DC:46:21:F0:39:E0:17:30:E8:4D:05:C7:FC:22:21:82:0
F:42

Certificate is to be certified until Apr 30 18:54:38 2012 GMT (365 days)
Sign the certificate? [y/n]:y


1 out of 1 certificate requests certified, commit? [y/n]y
Write out database with 1 new entries
Data Base Updated
unable to write 'random state'
Owner: EMAILADDRESS=lluismoran@uoc.edu, CN=CA, OU=ETIS, O=UOC, L=Barcelona,
ST=Barcelona, C=ES
Issuer: EMAILADDRESS=lluismoran@uoc.edu, CN=CA, OU=ETIS, O=UOC, L=Barcelona,
ST=Barcelona, C=ES
Serial number: b5cf82286eff9ded
Valid from: Sun May 01 20:48:55 CEST 2011 until: Fri Apr 29 20:48:55 CEST
2016
Certificate fingerprints:
    MD5:  81:41:83:E6:D7:F7:A5:AD:7B:AE:8A:5C:4B:A0:4B:DA
    SHA1: 42:18:48:D7:08:01:43:38:69:26:84:22:42:70:13:55:7E:23:EC:DA
    Signature algorithm name: SHA1withRSA
    Version: 3

Extensions:

#1: ObjectId: 2.5.29.14 Criticality=false
SubjectKeyIdentifier [
KeyIdentifier [
0000: 95 35 DC 46 21 F0 39 E0   17 30 E8 4D 05 C7 FC 22  .5.F!.9..0.M..."
0010: 21 82 0F 42                                        !..B
]
]

#2: ObjectId: 2.5.29.19 Criticality=false
BasicConstraints:[
  CA:true
  PathLen:2147483647
]

#3: ObjectId: 2.5.29.35 Criticality=false
AuthorityKeyIdentifier [
KeyIdentifier [
```

```
0000: 95 35 DC 46 21 F0 39 E0   17 30 E8 4D 05 C7 FC 22  .5.F!.9..0.M..."
0010: 21 82 0F 42                                        !..B
]

[EMAILADDRESS=lluismoran@uoc.edu, CN=CA, OU=ETIS, O=UOC, L=Barcelona, ST=Bar-
celona, C=ES]
SerialNumber: [    b5cf8228 6eff9ded]
]

Trust this certificate? [no]:  yes
Certificate was added to keystore
Owner: EMAILADDRESS=lluismoran@uoc.edu, CN=CA, OU=ETIS, O=UOC, L=Barcelona,
ST=Barcelona, C=ES
Issuer: EMAILADDRESS=lluismoran@uoc.edu, CN=CA, OU=ETIS, O=UOC, L=Barcelona,
ST=Barcelona, C=ES
Serial number: b5cf82286eff9ded
Valid from: Sun May 01 20:48:55 CEST 2011 until: Fri Apr 29 20:48:55 CEST
2016
Certificate fingerprints:
     MD5:  81:41:83:E6:D7:F7:A5:AD:7B:AE:8A:5C:4B:A0:4B:DA
     SHA1: 42:18:48:D7:08:01:43:38:69:26:84:22:42:70:13:55:7E:23:EC:DA
     Signature algorithm name: SHA1withRSA
     Version: 3

Extensions:

#1: ObjectId: 2.5.29.14 Criticality=false
SubjectKeyIdentifier [
KeyIdentifier [
0000: 95 35 DC 46 21 F0 39 E0   17 30 E8 4D 05 C7 FC 22  .5.F!.9..0.M..."
0010: 21 82 0F 42                                        !..B
]
]

#2: ObjectId: 2.5.29.19 Criticality=false
BasicConstraints:[
  CA:true
  PathLen:2147483647
]

#3: ObjectId: 2.5.29.35 Criticality=false
AuthorityKeyIdentifier [
KeyIdentifier [
0000: 95 35 DC 46 21 F0 39 E0   17 30 E8 4D 05 C7 FC 22  .5.F!.9..0.M..."
0010: 21 82 0F 42                                        !..B
]

[EMAILADDRESS=lluismoran@uoc.edu, CN=CA, OU=ETIS, O=UOC, L=Barcelona, ST=Bar-
celona, C=ES]
SerialNumber: [    b5cf8228 6eff9ded]
]

Trust this certificate? [no]:  yes
Certificate was added to keystore
Certificate reply was installed in keystore
S'ha creat el fitxer tomcatks.jks i el trustedca.jks a la carpeta tomcat.
Copieu-los a la carpeta d'instal·lació de Tomcat 7 i
seguiu les instruccions de la documentació per modificar
els fitxers de configuració.
```

Un cop finalitzada l'execució haurem de copiar els fitxers *tomcat.jks* i *trustedca.jks*, que podem trobar a la nova carpeta *tomcat*, cap a la carpeta d'instal·lació de Tomcat7.

Per últim hem de modificar el fitxer de configuració de Tomcat perquè arrenqui en SSL i només accepti connexions d'aquest tipus. Farem servir la implementació JSSE que és part del Java Runtime.

Modifiquem el fitxer *server.xml* de la carpeta *conf* de Tomcat

```
<!-- Define a SSL HTTP/1.1 Connector on port 8443
        This connector uses the JSSE configuration, when using APR,
        the connector should be using the OpenSSL style configuration
        described in the APR documentation -->

    <Connector protocol="org.apache.coyote.http11.Http11Protocol"
        port="8443" SSLEnabled="true"
        maxThreads="150" scheme="https" secure="true"
        keystoreFile="tomcatks.jks" keystorePass="3rt78jiK49"
        truststoreFile="trustedca.jks" truststorePass="3rt78jiK49"
        clientAuth="want" sslProtocol="TLS" />


   i modifiquem la línia:
  <!--APR library loader. Documentation at /docs/apr.html -->
  <Listener className="org.apache.catalina.core.AprLifecycleListener"
SSLEngine="off" />
```

Hem de tenir en compte que si modifiquem la contrasenya de la clau de tomcat abans de crear-la, també l'haurem de modificar al fitxer *server.xml* (clau keystorePass=”lanovacontrasenya” i truststorePass=”lanovacontrasenya”). Les dues claus fan servir la mateixa contrasenya

Un cop finalitzat aquest procés ja estem llestos per accedir a l'aplicació. Abans però, haurem d'iniciar (o reiniciar si ja estava corrent) el servidor Tomcat perquè carregui l'aplicació amb la nova configuració.

Abans d'accedir a l'adreça del servidor, per tal que funcioni correctament, haurem d'importar el certificat de la CA al nostre navegador o acceptar l'excepció que es produeixi, ja que la nostra CA no és reconeguda per cap autoritat de certificació de les incloses per defecte.

Ara ja podem executar l'aplicació des del navegador en qualsevol d'aquestes adreces:

https://localhost:8443/tfcpki/

http://localhost:8080/tfcpki/

Si fem servir la segona adreça, utilitzant una connexió no segura, serem re-dirigits a la primera. Quan es faci servir l'aplicació a través d'una connexió remota, haurem de canviar “localhost” per l'adreça IP o l'adreça de domini del servidor.

Un cop accedim a la interfície de l'aplicació ens haurem d'autenticar com a administrador des d'una connexió local (des del mateix servidor) i executar l'opció de Posada en marxa.

Per fer login com a administrador hem de fer servir el següent nom d'usuari i contrasenya, sempre que no els hàgim modificat en el fitxer de configuració d'usuari:

Usuari: **pkiadmin**

Contrasenya: **uoc**

Un cop tinguem accés, caldrà escollir l'opció de **Posada en marxa**, per tal de crear la base de dades i poder començar a fer servir l'aplicació.

(Nota: Les respostes de la consola poden variar en cada execució)