Project no. 027087

TENCompetence

Building the European Network for Lifelong Competence Development

Project acronym: Integrated Project TENCompetence

Thematic Priority: 2.4.10

ID3.21 Guidelines describing installation, configuration, maintenance and monitoring of the TENCompetence infrastructure

Due date of internal deliverable: 01-05-2009
Actual submission date: 08-07-2009

Start date of project: 01-12-2005 Duration: 4 years

LOGICA Version 1.0

Project co-funded by the European Commission within the Sixth Framework Programme (2002-2006)		
Dissemination Level		
PU	Public	X
PP	Restricted to other programme participants (including the Commission Services)	
RE	Restricted to a group specified by the consortium (including the Commission	
CO	Confidential, only for members of the consortium (including the Commission Services)	

Project internal Deliverable Report

ID3.21 Guidelines describing installation, configuration, maintenance and monitoring of the TENCompetence infrastructure

Work Package	WP 3 – Technical Design & Implementation of the Integrated System
Task	Task 1: Update the overall TENCompetence architectural design
Date of delivery	**Contractual**: 01-05-2009 **Actual**: 08-07-2009
Code name	ID3.21 **Version**: 1.0 Draft ☐ Final ☒
Type of deliverable	Report
Security (distribution level)	Public
Contributors	
Authors (Partner)	Jules Alberts (OUNL), Stef Heyenrath (LOGICA), Roy Cherian (University of Bolton)
Contact Person	Ruud Lemmers (LOGICA)
WP/Task responsible	WP3 / Ruud Lemmers (LOGICA)
EC Project Officer	Martin Májek
Abstract (for dissemination)	*Describes the software architecture of the "Amazon" release of the PCM to developers that need to use, extend or change the PCM server. The views from the 4+1 approach by Kruchten are used to describe the software architecture.*
Keywords List	Software installation, configuration

TENCompetence Project Coordination at: Open University of the Netherlands
Valkenburgerweg 177, 6419 AT Heerlen, The Netherlands
Tel: +31 45 5762605 – Fax: +31 45 5762800

Table of contents

1. Preface

This document describes how to install and configure the TENCompetence services and applications.

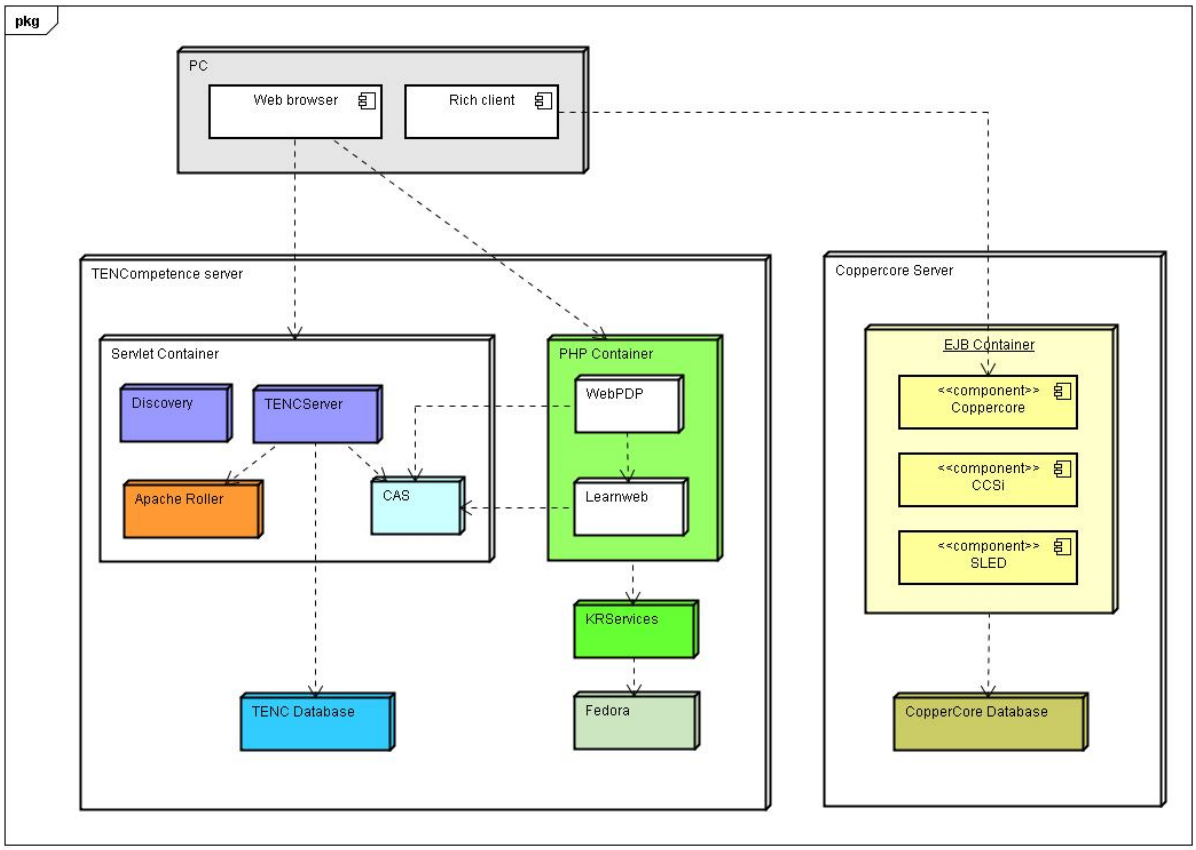
In this document a clear distinction will be made between the current environment located, setup and maintained in Sofia (in this document this server will be called CUR), and the future environment located in the Amazon EC2 cloud and setup and maintained by OUNL (called DEV). DEV contains (at this moment) less functionality and parts than CUR and is under heavy development. The missing functionality will gradually be incorporated or linked into DEV. In this document only functionality that is already present is described.

2. The TENCompetence infrastructure

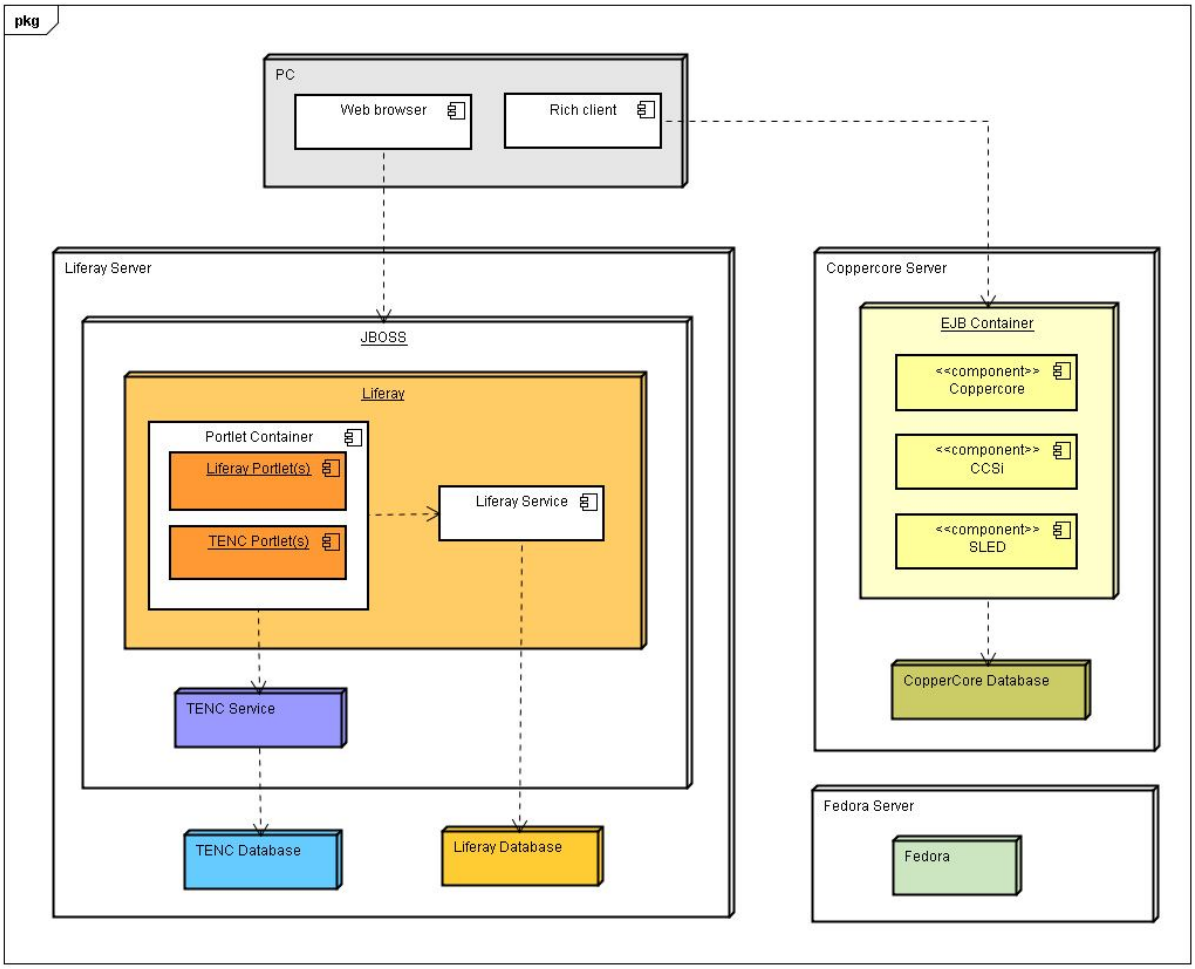
The TENCompetence infrastructure is schematically described in "Chapter 5 - Deployment view" from [5].

In this document, installation, configuration, maintenance and monitoring of the relevant parts of that chapter is described. The deployment schemes described in [5] are in *2.1* and *2.2*.

2.1. Infrastructure: current situation



2.2. Infrastructure: future situation



3. Installation and configuration of the CUR environment

3.1. TENC Server Installation and Configuration

This is described in the "TENC Server Installation and Configuration Manual" which is at [1].

3.1.1. Rich client (Recourse)

This is described by [4].

3.1.2. Learnweb server

These are not available yet, they will be published in internal deliverable *ID3.27 - Guidelines describing installation, configuration, maintenance and monitoring of the TENCompetence infrastructure* (month 47), which is the next version of this document describing the situation at that point in time.

3.1.3. Widget server

Liferay 5.2.3 with CCRT installation

This paragraph is aimed at system administrators setting up a Coppercore runtime environment (CCRT) within Liferay 5.2.3 plus JBoss 5.0.0. This software bundle can be downloaded from http://downloads.sourceforge.net/lportal/liferay-portal-jboss-tomcat-5.0-5.2.3.zip

The following is a list of configuration/installation steps that needs to be completed to get a working CopperCore run time (version 3.2) with CopperCore web service interfaces, SLED player, TENCompetence Link tool, TENCompetence Widget server, Link tool Liferay portlet and SLED Liferay portlet components.

1. Download the Liferay/JBoss bundle and unzip/install/configure as per Liferay documentation.
2. Check that the Liferay installation is working by starting liferay/login as administrator/normal user.
3. Download a zip file (file name liferayJBoss5.2.3_to_CCRT.zip, located at http://sourceforge.net/project/showfiles.php?group_id=159487&package_id=328381&release_id=692384 containing additional files for enhancing the Liferay bundle to act as CCRT.
4. Open this zip file and extract content/copy over the Liferay installation folder completed in Step 1. This should not result in any overwriting of existing Liferay/JBoss files as contents are Coppercore and its associated application jars and libraries.
5. If step 4 is completed without any overwrite, then a working CCRT installation is also part of Liferay.
6. The CCRT installation completed in Step 5 is configured for a Postgres database and this might require modification depending on the deployed environment.

7. To change this to MySQL or any other database, edit/remove the following files as appropriate:
   a. wookie-postgres-ds.xml (used by wookie widget server) – located in Liferay_Installation _Folder\jboss-tomcat-5.0.0\server\default\deploy
   b. coppercore-postgres-ds.xml (used by coppercore run time) – located in Liferay_Installation _Folder\jboss-tomcat-5.0.0\server\default\deploy
   c. hibernate.cfg.xml (used by normal linktool) – located in Liferay_Installation _Folder\jboss-tomcat-5.0.0\server\default\deploy\linktool.war\WEB-INF\classes
   d. hibernate.cfg.xml (used by linktool portlet) – located in Liferay_Installation _Folder\jboss-tomcat-5.0.0\server\default\deploy\linktool-portlet.war\WEB-INF\classes
   e. hibernate.cfg.xml (used by wookie widget server) – located in Liferay_Installation _Folder\jboss-tomcat-5.0.0\server\default\deploy\wookie.war\WEB-INF\classes
   f. In addition to the above, a database for coppercore (ld_publication) and wookie widget server (widgetdb) should exist in the database server used. SQL Scripts for creating the ld_publication database can be found in the source distribution of Coppercore (http://coppercore.sourceforge.net/downloads.shtml file name coppercore_sources_3.2.zip, sql scripts are in /coppercore/database folder).
   g. Similarly for widget server MySQL scripts can be found here: http://tencompetence.cvs.sourceforge.net/viewvc/tencompetence/wp6/org.tencompetence.widgetservice/scripts/
8. Start the database server(s) and then start Liferay by clicking the start_coppercore.bat file located in "Liferay_Installation _Folder". (not the run.bat file located in (Liferay_Installation _Folder)/bin)
9. This should start a new console in windows machines with Liferay portal start up logs as usual. There should not be any errors/exceptions when the server is deploying Coppercore components (coppercore, web service components, sled, linktool, wookie)
10. If deployments went smoothly, users should be able to access Liferay as usual and in addition access Coppercore components like SLED player, Linktool and Widget server via direct links (http://server_name:8080/linktool, http://server_name:8080/sled3 etc)
11. Two Coppercore based portlets should also be available in the Liferay portal environment. To add/configure these portlets, log in as administrator of the server. Select add components and in the list of categories shown, select "sample". There should a 'sled player' and 'link tool'. These are portlet versions of SLED player and link tool and can be used in the normal way by authenticated Liferay usersDs.

More information can be found at [2] (Widget server) and [3] (Coppercore installation).

3.1.4. Fedora Open Source Repository Software

Installation

To install the Fedora server on a Windows machine, take the following steps:

1) Download and install the Java Runtime Environment (JRE) 5.0 Update 6 (or a later 5.0 update or a 6.0 version) from http://java.sun.com/javase/downloads/index_jdk5.jsp
2) Download and install the MySQL 5.0.41 Community Server (or a later 5.0 update) from http://dev.mysql.com/downloads/mysql/5.0.html#downloads
3) Download Fedora Release 2.2.1 http://www.fedora.info/download

Prepare Environment Variables

The following environment variables must be correctly defined:

- *JAVA_HOME* This should point to the base directory of your Java installation.
- *FEDORA_HOME* This is the directory where Fedora will be installed, for example, C:\fedora.
- *PATH* This must include the Java and Fedora bin directories. For Windows, this will be %FEDORA_HOME%\server\bin, %FEDORA_HOME%\client\bin and usually %JAVA_HOME%\bin.
- *CATALINA_HOME* For Windows, this will be %FEDORA_HOME%\server\tomcat.

Installation Procedure MySQL Server Configuration

- Please note that the MySQL JDBC driver provided by the installer requires MySQL v3.23.x or higher. The MySQL commands listed below can be run within the mysql program, which may be invoked as follows: *mysql -u root -p*
- Create the database. For example, to create a database named fedora22, enter: *CREATE DATABASE fedora22;*
- Set username, password and permissions for the database. For example, to set the permissions for user fedoraAdmin with password fedoraAdmin on database fedora22, enter: *GRANT ALL ON fedora22.* TO fedoraAdmin@localhost IDENTIFIED BY 'fedoraAdmin'; GRANT ALL ON fedora22.* to fedoraAdmin@'%' IDENTIFIED BY 'fedoraAdmin';*

Fedora Configuration & Installation

(Full documentation is available at: http://www.fedora.info/download/2.2.1/userdocs/)

Download Fedora

The latest version of the software can be found at http://www.fedora.info/download/. There are two download options: the Fedora Installer and the source code distribution – download Fedora Installer.

Installing the Fedora Server

- To start the installer, change to the directory where you downloaded the installer and at a command prompt, enter: java -jar fedora-2.2.1-installer.jar
- In the dialogue menu enter custom installation type. Follow the steps and fill-in the appropriate values for parameters.

- On the step "*Authentication requirement for API-A*" choose option **false**.
- On the step "*SSL availability*" choose option **true**.
- On the step "*SSL required for API-A*" choose option **false**.

- On the step "*Database*" choose option mysql and fill-in appropriate values (see MySQL Server Configuration section)

Post installation configurations

The Fedora Server's configuration is chiefly governed by the Fedora Server Configuration File (fedora.fcfg) located at FEDORA_HOME\server\config\fedora.fcfg.

The Resource Index Module Configuration

The Resource Index module should be configured as follows:

```
<module role="fedora.server.resourceIndex.ResourceIndex"
class="fedora.server.resourceIndex.ResourceIndexModule">

    <param name="level" value="2"/>

    <param name="datastore" value="localKowariTriplestore"/>

    <param name="syncUpdates" value="true"/>

</module>
```

After modifying the resource index run %FEDORA_HOME%\server\bin\fedora-rebuild.bat and Choose “Rebuild the Resource Index” then “Yes”.

Persistent IDentifier namespaces Configuration

Fedora's support for Persistent IDentifier namespaces allows for PIDs to be generated from a pool of user specified namespaces that go beyond the default namespace specified in the fedora.fcfg pidNamespace property. It is important to remember that any custom namespace you wish to use outside of the default must be included in the retainPIDs parameter.

<param name="retainPIDs" value="**category user config tagging resource rating tag comment krsm-bdef-category krsm-bmech-category** demo test changeme fedora-bdef fedora-bmech tutorial">

<comment>Namespaces of PIDs to retain during the ingest process. When an object is ingested, Fedora normally allocates a unique PID within pidNamespace for it regardless of what the object says its PID is. This

option provides a way to override that behavior on a per-pid-namespace basis. If specified, this should be a space-delimited list of pid namespaces that will be accepted in the object as-is. Default value is"demo test".</comment>

</param>

Running Fedora Server

- Ensure that MySQL database server is running.
- `%FEDORA_HOME%\tomcat\bin\startup.bat`

Post Installation Configurations

Ingesting pre-defined bDef and bMech KRSM objects

Ingesting Behavior Definition Object (krsm-bdef-category:2) and Behavior Mechanism Object (krsm-bmech-category:2) used to define and implement the methods getParentCategory and getSubCategories.

- Select File/Ingest/One Object/From File... in the Fedora Administrator. This will bring up a file selection dialogue box as follows:
- Browse the file system to select the ingest file for the bDef object whose file name is *krsm-bdef-category_2.xml*. Since this ingest file is encoded as FOXML select the FOXML radio button as below:
- This will create the digital object with PID krsm-bdef-category:2 in your repository. This bDef defines two methods getParentCategory and getSubCategories.

Follow the same procedure to ingest a sample bMech object into the repository. This bMech represents a concrete implementation of the abstract service operations defined in the bDef krsm-bdef-category:2. Select the file krsm-bdef-category_2.xml. This will create the digital object with the PID currently *krsm-bmech-category:2*.

Use the same procedure to ingest User Behavior Definition Object (user_bdef.xml) and Behavior Mechanism Object (user_bmech.xml) to define and implement the method GetUserConfigs.

Ingesting KRSM root category object

Follow the steps from „Ingesting pre-defined bDef and bMech KRSM objects“ section and ingest `category_krsm-root.xml` file.

3.1.5. Web PDP

Introduction

This is the installation guide for Web PDP Client. It includes instructions for installing and configuring the client.

Installation Prerequirements

Required

PHP 5, PEAR, and Apache 2 should be installed on the machine you intend to use as a server for the Web PDP Client.

Additional requirements: enable mod_rewrite in Apache

- Locate the httpd.conf file (usually you will find it in a folder called conf, config or something along those lines)
- Inside the httpd.conf file uncomment the line LoadModule rewrite_module modules/mod_rewrite.so (remove the pound '#' sign from in front of the line) # to enable module `LoadModule rewrite_module modules/mod_rewrite.so`

Additional requirements: enable curl extension in PHP

Locate the php.ini file (usually you will find it in a folder called conf, config or something along those lines)
Inside the php.ini file uncomment the line containing php_curl extension (remove the ';' sign from in front of the line)

```
; to enable under windows
extension=php_curl.dll

; to enable under linux
;extension=php_curl.so
```

Installation

Download Web PDP Client

Create a local directory where Web PDP Client will be downloaded and installed. Change working directory to the newly created directory.

Anonymous CVS access

This project's CVS repository can be checked out through anonymous (pserver) CVS with the following instruction set. When prompted for a password for anonymous, simply press the Enter key.

```
cvs -d
:pserver:anonymous@tencompetence.cvs.sourceforge.net:/cvsroot/
tencompetence login (no password)

cvs -z3 -d
:pserver:anonymous@tencompetence.cvs.sourceforge.net:/cvsroot/
tencompetence co org.tencompetence.web.pdp
```

Prepare Environment Variables

Configure Apache name-based Virtual Host

You must have the name in DNS, resolving to your IP address, or nobody else will be able to see your web site. You can put entries in your hosts file for local testing, but that will work only from the machine with those hosts entries.

- Locate the hosts file (usually you will find it for Windows in a folder %SystemRoot%\system32\drivers\etc\ and for Linux in /etc)
  `127.0.0.1 localhost pdp.localhost`
- Locate apache virtual hosts configuration file (usually you will find it in a folder called conf, config or something along those lines) and add following lines (change paths according to you local file system, and directories created in previous steps)

```
<VirtualHost *:80>
ServerAdmin webmaster@pdp.localhost
ServerName pdp.localhost

DocumentRoot C:/development/tenc/org.tencompetence.web.pdp

<Directory "C:/development/tenc/org.tencompetence.web.pdp/">
Options Indexes FollowSymLinks
AllowOverride All
Order allow,deny
Allow from all

php_admin_value safe_mode 0
php_admin_value default_charset UTF-8
</Directory>

ErrorLog C:/xampp/apache/logs/pdp.localhost-error_log
CustomLog C:/xampp/apache/logs/pdp.localhost-access_log
</VirtualHost>
```

Configuring Web PDP Client

Change the url addresses according to your TENCompetence Server installation.

APP/config/tenc.config.php

```
$config = array();
Configure::write('roller.atom.endpoint',
'http://localhost:8080/roller/fp/feed/entries/atom');
Configure::write('roller.user.endpoint' ,
'http://localhost:8080/roller/roller-services/app');
Configure::write('roller.auth' , 'Basic');
```

APP/config/bootstrap.php

```
define('TENC_SERVER_URL', 'http://62.44.100.244:8080/TENCServer/');
define('MAX_UPLOAD_FILE_SIZE', 1048576);
define('HTTP_CODE_FORBIDDEN', 403);
define('DEFAULT_LANGUAGE', 'eng');
Configure::load('tenc.config' );
```

Running Web PDP

Restart Apache server and open a web browser with following URL http://pdp.localhost

References

- Apache Virtual Host documentation http://httpd.apache.org/docs/1.3/vhosts/ Apache Module mod_rewrite http://httpd.apache.Org/docs/2.0/mod/mod_rewrite.html
- PHP's cURL support http://php.net/curl.installation

4. Installation of the DEV environment

4.1. PC

The clientside can use a browser to access the functionality that is implemented on the Liferay server.

4.2. Liferay server

Note: All shell scripts in DEV should be tested thoroughly and provided with error handling before they are implemented in a live environment. Make sure these scripts are owned by root and have no access for group and others, they contain passwords!

4.2.1. Amazon EC2 setup

The DEV environment runs in the Amazon EC2 cloud. The process of creating servers ("instances"), volumes, elastic IP addresses, safety rules etc. is described in detail by Amazon: http://docs.amazonwebservices.com/AWSEC2/2008-02-01/GettingStartedGuide/.

The first steps is obtaining an AWS account on this address http://aws.amazon.com/ ("sign up now"), sign up for EC2 and get an X.509 certificate. Once you have this you can create server instances. For DEV, we use

- an Ubuntu 8.10 server image
- a 120 Gb volume that is mounted on the server to the directory /mnt/lifeRay1
- an elastic IP address that points to the instance
- a security group in order to open the ports 80 (HTTP), 8080 (JBoss), 22 (SSH. This should be another port on a live server, for instance 2343)

4.2.2. Ubuntu server work

Software update and extra software

- sudo apt-get update && sudo apt-get upgrade -y
- dpkg-reconfigure tzdata
- apt-get install sun-java6-bin sun-java6-jdk sun-java6-jre mysql-server vim unzip

Configure SSH access

- vim /etc/ssh/sshd_config: PubkeyAuthentication yes, PasswordAuthentication no
- leave and reenter ssh connection
- adduser jal (in our case)
- mkdir /home/jal/.ssh
- cp /root/.ssh/authorized_keys /home/jal/.ssh && chown -R jal:jal /home/jal/.ssh

Give the new user admin rights

- visudo: add # Members of the admin group may gain root privileges %admin ALL=(ALL) ALL
- groupadd admin
- usermod -a -G admin jal
- leave and reenter ssh connection, from now on as user jal

Mount the Amazon volume

- format the volume as ext3 and mount it on /mnt/lifeRay1
- mkdir /mnt/lifeRay1/download && cd /mnt/lifeRay1/download

Installing Liferay

Download and unzip liferay

- wget http://downloads.sourceforge.net/lportal/liferay-portal-jboss-tomcat-5.0-5.2.3.zip
- mkdir /mnt/lifeRay1/liferay-portal-5.2.3 && cd /mnt/lifeRay1/liferay-portal-5.2.3
- unzip liferay-portal-jboss-tomcat-5.0-5.2.3.zip
- ln -s /mnt/lifeRay1/liferay-portal-5.2.3 /usr/local/liferay

Install MySQL

- sudo apt-get install mysql-server, password ourMysqlPassword
- mysql -u root -p
- create schema lportal523;

Configure Liferay

Make the file /usr/local/liferay/portal-ext.properties with these contents

```
hot.undeploy.enabled=true
com.liferay.portal.servlet.filters.gzip.GZipFilter=false
com.liferay.portal.servlet.filters.header.HeaderFilter=true
jdbc.default.driverClassName=com.mysql.jdbc.Driver
jdbc.default.url=jdbc:mysql://localhost/lportal523?useUnicode=true&amp;char
acterEncoding=UTF-8&amp;useFastDateParsing=false
jdbc.default.username=root
jdbc.default.password=ourMysqlPassword
main.servlet.hosts.allowed=
tunnel.servlet.hosts.allowed=127.0.0.1,174.129.241.102,174.129.240.211,174.
129.241.81
tunnel.servlet.https.required=false
session.timeout=60
session.timeout.warning=1
dl.file.max.size=61440000
ig.image.max.size=61440000
ig.image.extensions=.gif,.jpeg,.jpg,.png,.GIF,.JPEG,.JPG,.PNG
mail.session.mail.store.protocol=smtp
mail.session.mail.transport.protocol=smtp
mail.session.mail.smtp.host=smtp.gmail.com
mail.session.mail.smtp.password=ourGmailPassword
mail.session.mail.smtp.user=ourGmailUser@gmail.com
mail.session.mail.smtp.port=465
mail.session.mail.smtp.auth=true
mail.session.mail.smtp.starttls.enable=true
```

```
mail.session.mail.smtp.socketFactory.class=javax.net.ssl.SSLSocketFactory
```

Information about these options can be found here:
http://docs.liferay.com/portal/5.2/official/liferay-administration-guide.pdf, in the chapter "Advanced Liferay Configuration".

Create the jboss user

Liferay will for security reasons be run under the user jboss.

- sudo useradd jboss
- sudo passwd jboss ourJbossPassword

- sudo chown -R jboss /usr/local/liferay
- sudo chown -R jboss /usr/local/liferay-portal-5.2.3

Create the JBoss logfile

Where JBoss will put console output. You can later follow output with the command `tail -f /usr/local/liferayjboss-tomcat-5.0.0/log/console.log`

- sudo mkdir /usr/local/liferayjboss-tomcat-5.0.0/log
- touch /usr/local/liferayjboss-tomcat-5.0.0/log/console.log

Remove the Liferay demo configuration

- sudo rm -r /usr/local/liferay/jboss-tomcat-5.0.0/server/default/deploy/sevencogs-hook.war
- sudo rm -r /usr/local/liferay/jboss-tomcat-5.0.0/server/default/deploy/sevencogs-theme.war

Create the jboss startup script

Put this script in /etc/init.d/jboss and sudo chmod +x /etc/init.d/jboss

```
#!/bin/bash

JBOSS_HOME=${JBOSS_HOME:-"/usr/local/liferay/jboss-tomcat-5.0.0"}
JBOSS_USER=${JBOSS_USER:-"jboss"}
JAVAPTH=${JAVAPTH:-"/usr/lib/jvm/java-6-sun/bin"}
JBOSS_CONF=${JBOSS_CONF:-"default"}
JBOSS_BIND_ADDR=${JBOSS_HOST:+"-b $JBOSS_HOST"}
JBOSSCP=${JBOSSCP:-
"$JBOSS_HOME/bin/shutdown.jar:$JBOSS_HOME/client/jnet.jar"}
RUNSCRIPT=run.sh
JBOSSSH=${JBOSSSH:-"$JBOSS_HOME/bin/$RUNSCRIPT -b 0.0.0.0 -c
$JBOSS_CONF $JBOSS_BIND_ADDR"}
LOGFILE="/usr/local/liferay/jboss-tomcat-5.0.0/log/console.log"

# needed for stopping jboss
PID_LIST=

if [ "$JBOSS_USER" = "RUNASIS" ]; then
```

```
  SUBIT=""
else
  SUBIT="su - $JBOSS_USER -c "
fi

if [ -n "$JBOSS_CONSOLE" -a ! -d "$JBOSS_CONSOLE" ]; then
  # ensure the file exists
  touch $JBOSS_CONSOLE
  if [ ! -z "$SUBIT" ]; then
    chown $JBOSS_USER $JBOSS_CONSOLE
  fi
fi

if [ -n "$JBOSS_CONSOLE" -a ! -f "$JBOSS_CONSOLE" ]; then
  echo "WARNING: location for saving console log invalid:
$JBOSS_CONSOLE"
  echo "WARNING: ignoring it and using /dev/null"
  JBOSS_CONSOLE="/dev/null"
fi

#define what will be done with the console log
JBOSS_CONSOLE=${JBOSS_CONSOLE:-"$JBOSS_HOME/log/console.log"}

JBOSS_CMD_START="cd $JBOSS_HOME/bin; $JBOSSSH"
JBOSS_CMD_STOP=${JBOSS_CMD_STOP:-"$JAVAPTH/java -classpath $JBOSSCP
org.jboss.Shutdown --shutdown"}

if [ -z "`echo $PATH | grep $JAVAPTH`" ]; then
  export PATH=$PATH:$JAVAPTH
fi

if [ ! -d "$JBOSS_HOME" ]; then
  echo JBOSS_HOME does not exist as a valid directory : $JBOSS_HOME
  exit 1
fi

find_pid() {
  PID_LIST=`ps aux | grep $RUNSCRIPT | grep $JBOSS_USER | grep -v
grep | sort -r | awk '{ print $2 }'`
}

case "$1" in
start)
  find_pid
  if [ "$PID_LIST" != "" ]; then
    echo An instance is already running at $PID_LIST
  else
    echo start openoffice
    soffice -headless -accept="socket,host=127.0.0.1,port=8100;urp;"
    echo start jboss
    cd $JBOSS_HOME/bin
    if [ -z "$SUBIT" ]; then
      eval $JBOSS_CMD_START >${JBOSS_CONSOLE} 2>&1 &
    else
      $SUBIT "$JBOSS_CMD_START >${JBOSS_CONSOLE} 2>&1 &"
    fi
```

```
  fi
  ;;
stop)
  find_pid
  echo $PID_LIST
  if [ "$PID_LIST" != "" ]; then
    for PID in `echo $PID_LIST`; do
      echo PID is $PID
      if [ -z "$SUBIT" ]; then
        kill -15 $PID
      else
        $SUBIT "kill -15 $PID"
      fi
     COUNT=0
      while [ $COUNT -lt 30 ]
      do
        EINDE_LOG=`tail -n 1 $LOGFILE`
        if [ "$EINDE_LOG" = "Halting VM" ]; then
          exit
        fi
        sleep 1
        let COUNT=COUNT+1
      done
      find_pid
      if [ "$PID" != "" ]; then
        echo "PID $PID did not stop in time and will be killed"
        if [ -z "$SUBIT" ]; then
          kill -9 $PID
        else
          $SUBIT "kill -9 $PID"
        fi
      fi
    done
  fi
  # stop openoffice
  killall soffice
  # mysqld_safe will sometimes eat CPU for no apperent reason. Kill
it
  # to be sure
  /etc/init.d/mysql stop
  killall mysqld_safe
  /etc/init.d/mysql start
  ;;
status)
  find_pid
  if [ "$PID_LIST" != "" ]; then
    echo "Running on $PID_LIST"
  else
    echo "Not running"
  fi
  ;;
restart)
  $0 stop
  $0 start
  ;;
*)
```

```
  echo "usage: $0 (start|stop|restart|help)"
esac
```

Let Liferay fill the database

- sudo /etc/init/d/jboss start
- sudo /etc/init/d/jboss stop

Reset the password for all users

In order to not have enabled demo accounts like "Bruno"

- mysql -u root -p // ourMysqlPassword
- -- reset the password for all users!
- use lportal52;
- update User_ set passwordEncrypted=0;
- update User_ set password_='some new password';

Test if Liferay is running

- sudo /etc/init/d/jboss start
- w3m http://127.0.0.1:8080

Open up the instance to the outside world from the AWS

In order to make Liferay accessible from outside of the server, you will have to follow these steps:

- go to the Amazon AWS console https://www.amazon.com/gp/aws/ssop/index.html?ie=UTF8&awstokenttl=43200&wreply=https%3A%2F%2Fconsole.aws.amazon.com%2Fec2%2Flogin!doAuthenticate&awsstrict=false&wtrealm=urn%3Aaws%3Asid%3A1M53AJJCH6XEAVC2CS82&awscbid=1QQFCEAYKJXP0J7S2T02&wa=wsignin1.1&awssig=zgU4JYlWrmCM8etdvLetyIftg1U%3D
- go to the security groups configuration
- edit the security group default
- add a custom connection tcp from 8080 - to 8080 - source ip 0.0.0.0/0
- create an elastic IP address and bind it to the Amazon instance
- request a domain name and bind it to the elastic IP address you just bound to the instance

Make sure OpenOffice is available

It happened that OpenOffice (that runs headless to make storing Liferay content as PDF etc. possible) disappeared. This script runs every minute to make sure OoO is running, and if not, start it again.

```
#!/bin/bash

SO_INSTANCE=`ps aux | grep soffice | grep -v grep`
JB_INSTANCE=`/etc/init.d/jboss status`
```

```
if [[ "$JB_INSTANCE" != "Not running" && "$SO_INSTANCE" = "" ]];
then
  logger soffice was not running, is started
  echo ooo is started
  soffice -headless -accept="socket,host=127.0.0.1,port=8100;urp;"
fi
```

4.2.3. Coppercore server

CopperCore is not running on the DEV environment yet. Instructions for installing and configuring a CopperCore environment are available at [6].

4.3. Configuration of the DEV environment

4.3.1. Liferay server

Theme deployment script

Every five minutes a script runs to build and deploy themes on the server. This way, theme developers (who should have a ssh account) can deploy a theme to the server without having the technical expertise normally needed to do this.

```
#!/bin/bash

BASEDIR=/usr/local/liferay/theme-deployer
SOURCEDIR=$BASEDIR/deploy
TARGETDIR=$BASEDIR/themes/celstec-theme/docroot/_diffs
DEPLOYDIR=/usr/local/liferay/deploy
FIRST=1
VERBOSE=1

function say {
  if [ "$VERBOSE" -eq "1" ]; then
    echo $1
    logger $1
  fi
}

pushd $SOURCEDIR || exit 1
for DEPLOYFILE in `ls *war`
do
  say "$DEPLOYFILE found, it will be built and installed"
  say "Wat one minute in case the theme upload is still running"
  sleep 1m
  if [ "$FIRST" -eq "1" ]; then
    TEMPDIR=`mktemp -d`
    FIRST=0
    say "tempdir is $TEMPDIR"
```

```
  fi
  say "$DEPLOYFILE is unpacked"
  unzip $DEPLOYFILE -d $TEMPDIR && rm $DEPLOYFILE
  # copy only these dirs to prevent an end-users fault to get
  # junk in the system
  for DIR in css images javascript templates WEB-INF
  do
    say "cp -r $TEMPDIR/$DIR $TARGETDIR"
    cp -r $TEMPDIR/$DIR $TARGETDIR
  done
  test -d $TEMPDIR && rm -rf $TEMPDIR
  say "$DEPLOYFILE is gedeployed"
  pushd $BASEDIR/themes/celstec-theme
  ant deploy
  chown -R jboss:jboss $DEPLOYDIR
  popd
done

say "remove the sourcedir"
rm -r $SOURCEDIR
mkdir $SOURCEDIR && chmod ugo+rwx $SOURCEDIR

popd
```

War and jar deployment script

This script will look into the home directory of certain users and deploy war and jar files it finds there. This will enable programmers to "drop in" their efforts without having to know the in-and-outs of the server. It runs every three minutes.

```
#!/bin/bash

LR_EXT_DIR=/usr/local/liferay/jboss-tomcat-
5.0.0/server/default/lib/ext
LR_WAR_DIR=/usr/local/liferay/deploy

for i in afi hma hvo jal lsc
do
  echo check $i
  # check the ext dir
  EXT_DIR=/home/$i/liferay_ext
  if [ -d $EXT_DIR ]; then
    CHANGED="false"
    cd $EXT_DIR
    for EXT_FILE in `ls`
    do
      CHANGED="true"
      mv -f $EXT_FILE $LR_EXT_DIR
      logger file $EXT_FILE will be moved
    done
    if [ "$CHANGED" = "true" ]; then
      logger jboss is restarted
      chown -R jboss:jboss $LR_EXT_DIR
      /etc/init.d/jboss stop
      sleep 1m
```

```
      /etc/init.d/jboss start
    else
      # for debugging
      echo nothing to do
      echo
    fi
  else
    # there wasn't any ext dir yet, make it and leave
    echo create dir $EXT_DIR
    mkdir $EXT_DIR
    chown -R $i $EXT_DIR
  fi

  # same for the war dir
  WAR_DIR=/home/$i/liferay_war
  if [ -d $WAR_DIR ]; then
    cd $WAR_DIR
    for WAR_FILE in `ls`
    do
      mv -f $WAR_FILE $LR_WAR_DIR
      logger file $WAR_FILE is moved
    done
    chown -R jboss:jboss $LR_WAR_DIR
  else
    # there wasn't any ext dir yet, make it and leave
    echo create dir $WAR_DIR
    mkdir $WAR_DIR
    chown -R $i $WAR_DIR
  fi

done
```

4.3.2. Coppercore server

CopperCore is not running on the DEV environment yet. Instructions for installing and configuring a CopperCore environment are available at [6].

4.4. Maintenance of the DEV environment

4.4.1. Liferay server

Hourly backup of the database

During office hours (in crontab `43 8-19 * * 1-5 root /usr/local/sbin/db_dump.sh >> /var/log/backup`) a dump of MySQL is made with this script

```
#!/bin/bash

if [[ $EUID -ne 0 ]]; then
  echo run this script as root
  exit 1
fi

clear
TARGET_BASE=/mnt/lifeRay1/backups
MYSQLUSER=root
MYSQLPASSWORD=ourMysqlPassword

function makebackupdir {
  YEAR=`date +%Y`
  MONTH=`date +%m`
  DAY=`date +%d`
  test -d $TARGET_BASE/$YEAR || mkdir $TARGET_BASE/$YEAR || exit 1
  test -d $TARGET_BASE/$YEAR/$MONTH || mkdir
$TARGET_BASE/$YEAR/$MONTH || exit 1
  test -d $TARGET_BASE/$YEAR/$MONTH/$DAY || mkdir
$TARGET_BASE/$YEAR/$MONTH/$DAY || exit 1
  TARGET_DIR=$TARGET_BASE/$YEAR/$MONTH/$DAY
  chmod go-rwx $TARGET_DIR
}

function say {
  TIME=`date +%Y%m%d-%T`
  echo "=== HB $TIME - $1"
}

makebackupdir
cd $TARGET_DIR || exit 1
say "backup mysql"
TIME=`date +%Y%m%d-%T`
mysqldump --all-databases --create-options --lock-all-tables \
  --user=$MYSQLUSER  --password=$MYSQLPASSWORD >| $TIME.mysql.dump

if [ "$?" -ne "0" ]; then
  rm -r $TIME.mysql.dump
fi
```

Nightly backup of the system

Every night this script makes a backup of the vital parts of the server. This backup should suffice to create a new system from scratch, given a fresh Ubuntu image on Amazon. Note that this approach is by far insufficient for a live environment. The backups are stored on the Amazon volume, ideally these backups would be transferred to another, local, medium in case Amazon would have problems. So far no problems have occured though. Also at the moment we make (Amazon) snapshots, on an irregular basis.

Another solution, which would make a quick restore to the Amazon cloud possible, would be

to create an "AMI". This is a kind of an image of an Amazon instance (not the volume or the rest of the settings) that would make a quick restore of a complete instance possible. More info here: http://developer.amazonwebservices.com/connect/entry.jspa?externalID=368.

```
#!/bin/bash

if [[ $EUID -ne 0 ]]; then
  echo run this script as root
  exit 1
fi

clear
TARGET_BASE=/mnt/lifeRay1/backups
EXCLUDE=/usr/local/sbin/backup_exclude

function makebackupdir {
  YEAR=`date +%Y`
  MONTH=`date +%m`
  DAY=`date +%d`
  test -d $TARGET_BASE/$YEAR || mkdir $TARGET_BASE/$YEAR || exit 1
  test -d $TARGET_BASE/$YEAR/$MONTH || mkdir
$TARGET_BASE/$YEAR/$MONTH || exit 1
  test -d $TARGET_BASE/$YEAR/$MONTH/$DAY || mkdir
$TARGET_BASE/$YEAR/$MONTH/$DAY || exit 1
  TARGET_DIR=$TARGET_BASE/$YEAR/$MONTH/$DAY
  chmod go-rwx $TARGET_DIR
}

function say {
  TIME=`date +%Y%m%d-%T`
  echo "=== NB $TIME - $1"
}

function do_it {
  say "$1 backuppen"
  DOEL=`basename $1`.tar.bz2
  CMD="tar -c -P -v -j -X $EXCLUDE -f $DOEL $1"
  nice -n 10 $CMD
}

makebackupdir
cd $TARGET_DIR || exit 1

say "stop liferay for backup"
/etc/init.d/jboss stop

VER=`lsb_release -si`
TIME=`date +%Y%m%d-%T`
say "remember package selection"
case "$VER" in
"Ubuntu")
  dpkg --get-selections > $TARGET_DIR/$TIME.package_selection.txt
```

System software updates

Every night `apt-get update && apt-get upgrade -y` is run to make sure the system is up to date. **Note: this should be done controlled (not by crontab but by a human after reading upgrade notes and testing) in a live environment.**

Script to check for zombies

In case processes should become zombie, this script is cronned every seven minutes to (try and) kill them

```
#!/bin/bash
# Zombie processes killing script.
# Must be run under root.
case "$1" in
--admin)

stat=ps ax | awk '{print $1}' | grep -v "PID" | xargs -n 1 ps lOp |
grep -v "UID" | awk '{print"pid: "$3" *** parent_pid: "$4" ***
status: "$10" *** process: "$13}' | grep ": Z"

if ((${#stat} > 0));then

echo zombie processes found:

echo .
ps ax | awk '{print $1}' | grep -v "PID" | xargs -n 1 ps lOp | grep
-v "UID" | awk '{print"pid: "$3" *** parent_pid: "$4" *** status:
"$10" *** process: "$13}' | grep ": Z"
echo -n "Kill zombies? [y/n]: "
read keyb
if [ $keyb == 'y' ];then

echo killing zombies..
ps ax | awk '{print $1}' | grep -v "PID" | xargs -n 1 ps lOp | grep
-v "UID" | awk '{print$4" status:"$10}' | grep "status:Z" | awk
'{print $1}' | xargs -n 1 kill -9

fi

else

echo no zombies found!

fi

;;
--cron)

stat=ps ax | awk '{print $1}' | grep -v "PID" | xargs -n 1 ps lOp |
grep -v "UID" | awk '{print"pid: "$3" *** parent_pid: "$4" ***
status: "$10" *** process: "$13}' | grep ": Z"
```

```
if ((${#stat} > 0));then
ps ax | awk '{print $1}' | grep -v "PID" | xargs -n 1 ps lOp | grep
-v "UID" | awk '{print$4" status:"$10}' | grep "status:Z" | awk
'{print $1}' | xargs -n 1 kill -9

echo date": killed some zombie proceses!" >> /var/log/zombies.log
fi

;;
*) echo 'usage: zombies {--cron|--admin}'
;;
esac
exit 0
```

4.4.2. Coppercore server

There is no CopperCore server running in the DEV environment yet. Maintenance for CopperCore will be arranged when installing it.

4.5. Monitoring of the DEV environment

4.5.1. Liferay server

In a live environment much more monitoring should be implemented. This could be done with tools like Nagios / Cacti, plus there should be availability checking done from outside the Amazon instance to see if it and it's services are still available. For now the development server only does a regular check to see if the server is running out of disk space.

Disk space check

Every hour a script runs that checks if the server instance and the volume have a reasonable amount of free space left. If not, a mail is sent out to inform the maintainer of the server.

```
#!/bin/bash

LOCATIONS='/ /mnt/lifeRay1'
TO=maintainer@tc.eu # YMWV

for LOCATION in $LOCATIONS
do
  PERCENT_USED=`df $LOCATION|awk '{print $5}'|grep -v "Use%"|cut -f1
-d%`
  logger $LOCATION is using $PERCENT_USED procent of its space
  if [ "$PERCENT_USED" -gt "75" ]; then
```

```
    # assume the environment variable $EC2_INSTANCE is set
    /usr/local/sbin/mail.sh $TO \
      "Diskspace shortage on server $EC2_INSTANCE" \
      "$PERCENT_USED percent of disk $LOCATION is used"
  fi
done
```

This is the utility script that sends the mail

```
#!/bin/bash

TO=$1
SUBJECT=$2
MSG=$3

if [[ $EUID -ne 0 ]]; then
  echo run this script as root
  exit 1
fi

if [[ "$TO" = "" || "$MSG" = "" ]]; then
  echo Usage: $0 to_address \"subject\" \"message\"
  exit 1
fi

mailx -r celstec.ounl@gmail.com \
  -s "$SUBJECT" \
  -S smtp=smtp.gmail.com \
  -S smtp-use-starttls \
  -S smtp-auth=login \
  -S smtp-auth-user=celstec.ounl@gmail.com \
  -S smtp-auth-password=ourGmailPassword \
  $TO <<EOT
$MSG
EOT
```

References

[1] *TENCompetence Server Installation and Configuration Manual*, http://www.partners.tencompetence.org/file.php/7/Binaries/TENCServer/2009-05-06_v3.0beta6/tencs_documentation.zip

[2] *LD Runtime documentation location*, http://www.tencompetence.org/ldruntime/

[3] *LD Runtime installation notes,* http://www.tencompetence.org/ldruntime/resources/installing.doc

[4] Recourse *documentation location*, http://www.tencompetence.org/ldauthor/

[5] ID3.18 Architecture Design Document

[6] CopperCore – The IMS Learning Design Engine, http://www.coppercore.org

Links checked on 08-07-2009.