**Software Pre-requisites**

For the course, the lecturers created a Virtual Machine in advance that has the following software installed.

1. Ubuntu 14.04 Desktop edition 64-bit  
     
   Default userid / password = ox-soa/ox-soa
2. Install vm-ware tools (see vmware docs)
3. Do an Ubuntu update  
   sudo apt-get update  
   sudo apt-get upgrade
4. Install gksudo (for Eclipse)

Sudo apt-get install gksu

1. Java Development Kit JDK 1.7, Oracle Edition  
   sudo apt-get install default-jdk
2. Set JAVA\_HOME in .bashrc  
   export JAVA\_HOME=/usr/lib/jvm/java-7-openjdk-amd64
3. Also CXF\_HOME and Path in .bashrc

export CXF\_HOME=~/servers/apache-cxf-2.7.13  
export PATH=$PATH:~/servers/apache-cxf-2.7.13/bin

1. Apache Maven 3.1.3 or later\*

Apache Ant 1.9.3 or later\*  
Curl 7.35.0 or later\*

Tree\*  
  
sudo apt-get install ant maven curl tree

1. Google Chromium\* (or Chrome)  
   sudo apt-get install chromium
2. Google Chrome Advanced REST Client extension  
   <https://chrome.google.com/webstore/detail/advanced-rest-client/hgmloofddffdnphfgcellkdfbfbjeloo>   
   (to be installed from Chromium)
3. SOAPUI 5.0.0 or later\*  
   <http://sourceforge.net/projects/soapui/files/soapui/5.0.0/SoapUI-x64-5.0.0.sh/download>

chmod +x SoapUI-x64-5.0.0.sh

./SoapUI-x64-5.0.0.sh

1. Some extra text editors  
   sudo apt-get install cream leafpad
2. WSO2 Developer Studio 3.7.0\*  
   <http://wso2.com/products/developer-studio/>
3. This was un-zipped into the ~/eclipse directory  
     
   Then we did   
   sudo mv eclipse /opt/  
   sudo ln –s /opt/eclipse/eclipse /usr/bin/eclipse  
   Start eclipse from the command line and then “Lock to Launcher”
4. Before you import any maven project, you do need to let Eclipse know where your Maven is installed.   
   You can do this manually in Eclipse by adding the M2\_REPO variable, but there is also a command line tool for this:  
   mvn -Declipse.workspace=/home/ox-soa/workspace eclipse:add-maven-repo
5. Also need to set cxf.home in the ant config in Eclipse as a property.
6. Also we need already downloaded the following links into a common downloads folder:  
   Apache Tomcat 7.0.57: http://tomcat.apache.org/download-70.cgi#7.0.57

Unzip tomcat into the servers directory. Rename to tomcat.   
chmod +x bin/\*.sh

1. Apache CXF 2.7.13\*: <http://cxf.apache.org/download.html>   
   Unzip cxf into the servers directory. (Not 3.0.2 because of Eclipse Kepler issues)
2. The following servers were downloaded and unzipped into the ~/servers/ directory:

WSO2 App Server 5.2.1\*: <http://wso2.com/products/application-server/>   
WSO2 ESB 4.8.1\*: <http://wso2.com/products/enterprise-service-bus/> (Download “Binary”)  
WSO2 Governance Registry 4.6.0 \*: <http://wso2.com/products/governance-registry/> (Download binary)  
WSO2 API Manager 1.7.0\*: <http://wso2.com/products/api-manager/> (Download Binary)  
WSO2 Business Activity Monitor 2.4.1\*: <http://wso2.com/products/business-activity-monitor/> (Download Binary)  
WSO2 Business Process Server 3.2.0 \*: <http://wso2.com/products/business-process-server/> (Download Binary)

1. Unzipped Apache tcpmon into servers, renamed the directory to **tcpmon** and did chmod +x tcpmon.sh
2. Changed the port offsets in the servers as follows:

|  |  |  |  |
| --- | --- | --- | --- |
| Server | Offset | Directory | Admin Port / web |
| Tomcat | NA | ~/servers/tomcat | <http://localhost:8080> |
| App Server | 0 | ~/servers/wso2as-5.2.1 | <https://localhost:9443> |
| Enterprise Service Bus | 1 | ~/servers/wso2esb-4.8.1 | <https://localhost:9444> |
| Governance Registry | 2 | ~/servers/wso2greg-4.6.0 | <https://localhost:9445> |
| Business Process Server | 3 | ~/servers/wso2bps-3.2.0 | <https://localhost:9446> |
| API Manager | 4 | ~/servers/wso2am-1.7.0 | <https://localhost:9447> |
| Business Activity Monitor | 5 | ~/servers/wso2bam-2.4.1 | <https://localhost:9448> |

1. Download the code from Github into Downloads.
2. Download the keys (having maybe updated them!?) into backup\_keys
3. Git clone into a repo directory
4. Configure BAM to talk to API Mgr (or is it the other way round?)
5. Install node.js and npm   
   sudo apt-get install node.js npm
6. sudo nano /etc/security/limits.conf

#<domain> <type> <item> <value>

#

\* soft nproc 60000

\* hard nofile 10000

\* soft nofile 10000

28) Add the following .pam\_environment  
MOZILLA\_FIVE\_HOME=/usr/lib/Mozilla  
LD\_LIBRARY\_PATH=${MOZILLA\_FIVE\_HOME}:${LD\_LIBRARY\_PATH}

29) Install Camunda BPMN modeler into Eclipse using <http://camunda.org/release/camunda-modeler/update-sites/kepler/latest/site/>

30) Download <http://camunda.org/release/camunda-bpm/tomcat/7.3/camunda-bpm-tomcat-7.3.0.zip>

31) In ~/servers

md camunda

cd camunda

unzip ~/Downloads/camunda-bpm-tomcat-7.3.0.zip

32) In the Camunda tomcat conf directory, in server.xml edit the port from 8080 to 8090.

33) Download coffee-approval.zip from <https://github.com/pzfreo/ox-soa/blob/master/lab-exercises/code/coffee-approval.zip?raw=true>

Tidy up before handing to students:

1. Re-install any servers and edit offset.
2. Delete all Eclipse projects and make sure workspace is empty
3. Remove Eclipse Servers
4. Remove CXF environment
5. Remove any Tomcat webapps (or re-install and rename)
6. Delete any generated code projects
7. Check the keyboard setting

\* All the items marked \* are Open Source. This entire course can be done using 100% open source.