Example CSAR “Moodle”
For OpenTOSCA Release v1.1
OpenTOSCA Container
How to use Moodle.csar
Moodle Example

- **Moodle** is an open-source course/school/learning management system based on an LAMP-stack on Amazon EC2

- There are two ways to prepare and run Moodle.csar:
  - (1) **Moodle + Vinothek**, i.e., the self service way
    - Adaptation & CSAR deployment is done by admin role (AdminUI)
    - Instantiation is done by end user in self-service portal (Vinothek)
  - (2) **Moodle + soapUI**
    - CSAR deployment & instantiation (done by admin role)

- **Troubleshooting tips if something does not work**
Required Information for (1) and (2)

To run Moodle.csar on Amazon EC2 you need the following information from your AWS Account:

- Region to run (e.g., “ec2.eu-west-1.amazonaws.com”)
- Ubuntu Linux AMI available in this region
  - Tested with Ubuntu Server 13.10 64bit
  - E.g., “ami-ec50a19b” for eu-west-1
  - Locate via [http://cloud-images.ubuntu.com/locator/ec2/](http://cloud-images.ubuntu.com/locator/ec2/) or the Amazon EC2 Instance Creator Wizard
- Your account’s access and secret key
- Security group (same region!): All TCP ports open
- Key name and certificate of a key pair in the respective region
(1) Moodle + Vinothek: Preparations

**Instantiate Moodle using Vinothek self-service UI:**

- **(1.1) Vinothek requires the AWS information in the predefined build plan input message**
  
  - Therefore, adapt the plan input messages accordingly by adding your AWS information to the build plan messages `Moodle.csar/SELFSERVICE-Metadata/plan.input.default.xml`
  
  - The CSAR is a ZIP archive which can be extracted and edited using standard ZIP tools. To adapt Moodle.car unzip it, modify the respective files, and ZIP it again. Check that the folder structure is conserved.
(1.2) Moodle + Vinothek

- (1.2) Deploy Moodle.csar to container
  - Go to http://<CONTAINER-HOST>:8080/
    and click on “Administrative UI”
  - “Upload new CSAR” (Button lower left)
(1.3) Moodle + Vinothek

(1.3) Instantiate using Vinothek

- Go to http://<CONTAINER-HOST>:8080/ and click on “Vinothek Self-Service Portal”
- Click on Moodle application listed there to open detail view
- Click “Start Instance”
- Approx. 4 min later: Click “play” button to open Moodle instance

Do not forget to shutdown the created instances on Amazon! 😊
(2) Moodle + soapUI

Instantiate Moodle by invoking the build plan manually with SOAPui:

- No Adaptation of Moodle.csar required
- (2.1) Deploy Moodle.csar to container
  - Go to http://<CONTAINER-HOST>:8080/ and click on “Administrative UI”
  - “Upload new CSAR” (Button lower left)
(2.2) Start build plan using soapUI

- File → New soapUI Project
- Initial WSDL: http://<CONTAINER-HOST>:9763/services/MoodleBuildPlanService?wsdl
- Open: MoodleBuildPlanBinding → initiate → Request1
- Fill in your data (using the template on the following slide)
- Submit request by pressing the green play button

Do not forget to shutdown the created instances on Amazon! 😊
(2.2) Moodle + soapUI (SOAP Message template)

Replace everything in red font:

```xml
<soapenv:Envelope xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/
xmlns:org="http://www.opentosca.org/examples/Moodle(BuildPlan">
  <soapenv:Header/>
  <soapenv:Body>
    <org:MoodleBuildPlanRequest>
      <org:region>AW-S-REGION-NAME (e.g., "ec2.eu-west-1.amazonaws.com")</org:region>
      <org:securityGroup>AW-S-SECURITY-GROUP-NAME (create one with all TCP ports open)</org:securityGroup>
      <org:keyPairName>AW-S-KEY-PAIR-NAME</org:keyPairName>
      <org:sshKey>-----BEGIN RSA PRIVATE KEY-----
      ...
      AWS-SSH-KEY
      (Ensure that all line breaks are conserved and no white spaces at the beginning of the lines are added!)
      ...
      abcdefghijklmnopqrstuvwxyz0123456789=
      ------END RSA PRIVATE KEY-----</org:sshKey>
      <org:ami>AW-S-AMI-ID
      (e.g., “ami-ec50a19b” for eu-west-1; Ubuntu Server 13.10 64bit in *your* region)</org:ami>
      <org:instanceType>t1.micro</org:instanceType>
      <org:accessKey>AW-S-ACCESS-KEY</org:accessKey>
      <org:secretKey>AW-S-SECRET-KEY</org:secretKey>
      <org:csarName>Moodle.csar</org:csarName>
      <org:callbackUrl>http://example.org</org:callbackUrl>
      <org:CorrelationID>randomId123</org:CorrelationID>
    </org:MoodleBuildPlanRequest>
  </soapenv:Body>
</soapenv:Envelope>
```
(2.3) Moodle + soapUI

(2.3) Receiving the result of the build plan is a little bit tricky 😊

- The build plan is *asynchronous*, i.e., the request returns immediately and the build plan uses the callback URL in the plan input message to return its results.

- To receive the reply create a soapUI MockService
  - Right click “MoodleBuildPlanCallbackBinding”
  - Select “Generate MockService”
  - Click ok, the window will print all responses it receives

- Pass the MockService URL as callbackURL in the build plan input message

- **However**, if you have the container running on Amazon and soapUI on a machine not accessible from the web, the build plan will not be able to send you the reply!

- **However**, it is still possible to see the deployed application
  - After the build plan finished (~10 min) go to the Amazon EC2 console and copy the public DNS address of the second t1.micro instance created by the build plan
  - Open: http://<publicDNS>/moodle to see the deployed Moodle

- All the heavy lifting done here is wrapped by the Vinothek, so we recommend using the Vinothek when not debugging a specific problem with your build plan!
Troubleshooting Tipps

- Is the management plan deployed? (see BPS web console linked from the OpenTOSCA root page)
- After starting an plan instance from Vinothek, is there a process instance in WSO2 BPS?
  - If yes, what’s the status of the process instance, did it fail?
  - If yes, check BPS log for more details. You can find it somewhere here (not sure about the exact path): /wso2bps/repository/log/...
- Does the keyPairName you’re using exist in the EC2 region you’re using?
- Is an Amazon EC2 instance created, i.e., listed in your AWS Console?
  - If yes, are you able to connect onto this machine using SSH (user “ec2-user”) and the credentials you provide in the plan.input.default.xml?
  - If no, check Tomcat Log (/var/log/tomcat7/catalina.out) for error messages of the Amazon Implementation Artifact trying to launch an EC2 instance. If you AWS credentials or configuration is wrong, you’ll see this in the Tomcat log.
- Have you been able to start the build plan using SOAPui?