

## Case Study

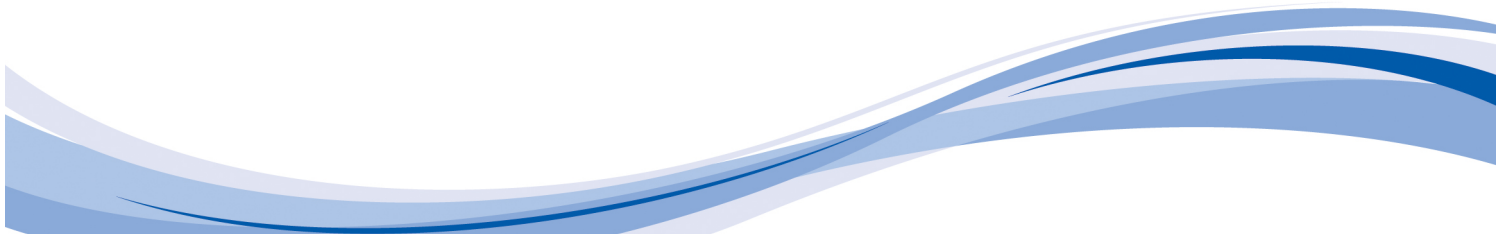
### Objective

The objective of the case study is to work in groups or pairs to build up an end-to-end tool chain that will build, containerise and deploy some software. There are no step-by-step instructions for the case study as it is up to you how you design and build your system. You should work in an agile manner - plan what you will do and take ownership of different parts of the project to work on.

You will have access to AWS to spin up servers. Please ensure that you terminate anything at the end of the day.

### Requirements

- Create a gitlab server to host your repositories
- Create a new project based on the simple-scalatra maven archetype
- Host your project on your gitlab server
- Setup a Jenkins server and use it to build your new scalatra project
  - We want the build to be automated, so set it up to build every time a change is pushed to git
  - You will need to use the gitlab plugin, rather than the bitbucket one for this!
- Install the docker plugin for Jenkins and write a docker file to build a container for your project
  - The docker container should have a tomcat server
  - It should copy the built .war file to the tomcat webapps directory so that it is launched
  - It would be nice to have a username and password for the manager GUI in tomcat as well, so add a tomcat-users.xml file to be copied into tomcat as well
- When the docker file is created, push the container to the docker hub
- Create a puppet master server
- Write a module for your hello-scalatra Docker container
- Create a group for docker machines, using regular expressions to automatically add new nodes to this group. Pin the module to this group.



- Create a nagios server and use it to monitor the other servers you have running
  - You can either build it from source code or install via the repository
  - Your servers to monitor are Gitlab, Jenkins, puppet master and the agent

### If you have more time

- We currently need to manually accept the node registration on the puppet master. Have a look at [https://docs.puppetlabs.com/puppet/latest/reference/ssl\\_autosign.html](https://docs.puppetlabs.com/puppet/latest/reference/ssl_autosign.html) to see if there is way to automate this
- We needed to create the agent manually in aws. Have a look at either using puppet, Jenkins, Docker or even the command line to start a new node for you to automate the last step

The following links may help, but they may not be the best way!

- Jenkins EC2 Plugin: <https://wiki.jenkins-ci.org/display/JENKINS/Amazon+EC2+Plugin>
- Docker: <https://docs.docker.com/machine/>
- Puppet: <https://puppetlabs.com/solutions/aws>
- We are using the main docker hub for our projects. See if you can create your own docker registry to host your containers and pull from
- If you get this one up and running there is a PHP and MySQL based project available at <https://bitbucket.org/qatrainingauthors/eventspj> - see if you can get that project pulled in by Jenkins, Dockerised and hosted. There are no tests to run or compilation requirements, so all Jenkins will need to do is set off the Docker process.
  - You will need to have two linked docker containers, one with the database and one with apache (or nginx)
  - Have a look at docker-compose to see how to set off and control more than one container at once
  - Create a puppet manifest and assign it to an agent to run the project
  - Monitor everything with nagios

