COGSI - Configuração e Gestão de Sistemas

Mestrado em Engenharia Informática, Ramo Sistemas Computacionais

Lecture 12.1

P2.1 - Integration

Alexandre Bragança atb@isep.ipp.pt

Dep. de Engenharia Informática – ISEP

2018/2019

P2.1: Overview

- **Start Date:** 20, May
- **End Date:** 3, June (no commits after this date!)
- **Development Repository:** Your individual repository
  - Create Issue(s) for your work
  - Expected several commits (at least 1 for each lab class!)
  - You should commit to the repository only files that you created or edited (e.g., do not commit the nagios directory!)
  - Documentation should be provided **only in the readme.md file related to the assignment!**
- **Notes regarding Evaluation of P2:**
  - P2.1 stands for 25% of the final grade (no minimum score)
  - P2.2 stands also for 25% of the final grade (no minimum score) and is to be developed in the lab class of the week 3 to 7 of June

P2.1: Topic

- The topic of this assignment is the **integration** of all the tools and technologies used during P1.
- The overall goal is to simulate an integrated "devops" scenario.
- One main concern of this assignment (that is also core in "devops") is to treat **infrastructure as code**.
  - So, you should try as much as possible to achieve this goal;
  - The artifacts in your repository should be sufficient to build the application and also the infrastructure that will run it.
- For this assignment we will also be using **Jenkins** and **Nexus** (see slides from the previous week).
- The idea is to implement a pipeline for the continuous integration of the **TODD** application.
- This pipeline should also include the continuous delivery of the application by publishing it and eventually deploying it.
- A small local network should be simulated (that includes the hosts where we will be deploying TODD)
- The hosts and services (i.e., TODD) should be monitored.

P2.1: Specific Concerns (1/2)

Continous Integration

- You should use Jenkins to implement a pipeline to build the TODD application (located in your individual repository)
- You may follow the example available in the TODD repository (`https://bitbucket.org/mei-isep/todd`).

Continous Delivery

- Successful builds of the application should result in the application being archived in the Nexus artifact repository (see slides from the previous week)
- It should also be possible to deploy the application to, at least, two different hosts in the local network.
- This should be done using Ansible
  - You should try to include a stage in the pipeline that, upon user confirmation, executes the Ansible deployment of the application

Network

- All the hosts should be connected
  - You may reuse the assignment of GNS3 or implement the local network using the features of Docker or Vagrant
- The network should include two hosts:
  - One with the TODD application
  - One with Nagios and the TODD application
- **Note:** You are free to install and setup all the other software (e.g., Ansible, Jenkins, Nexus, etc.) as you see more fit.

P2.1: Specific Concerns (2/2)

Virtualization

- To simulate hosts you may use Docker or Vagrant.
- It is not mandatory to use GNS3 (but you are free to use it)

Configuration Management

- To deploy the TODD application you must use Ansible
- Ansible should make sure the requirements for the application are present (e.g., java)
- Ansible should also apply the necessary configuration so that Nagios is able to monitor the TODD server status and an event handler to try to automatically restart it in case of failure

Monitoring

- Nagios should monitor TODD (in the two hosts)
- You should use JMX to monitor TODD

P2.1: Alternative

- For this assignment you may explore alternatives on how to implement the "continuous deployment" aspects.
- For instance, using different approaches in the pipeline; using different artifact repositories; using containers to package the application, etc.

P2.1: Technical Report and Included Artifacts

You should produce a **technical report** documenting your assignment.

- The technical report **must be produced** in the **readme.md** file located in the repository folder related to P2.1 (e.g., 1133224-maria-ferreira/p2.1/)
- The report should include:
  - The Analysis of the Problem
  - The Design of your Solution
    - Present an overview of the tools (e.g., software used, major concepts, major processes, architecture of the tools)
    - Present an overview of the solution (e.g., the architecture and major configurations required)
  - The Steps required to Reproduce your Solution (it should include references/links to configuration files, scripts or code included in the same folder of the repository)
- You may also include:
  - Justification of Design Options
  - Analysis of the Alternative
  - The Steps required to Reproduce the Alternative