

JePL demo (Jenkins Pipeline Library)

Pablo Orviz <orviz@ifca.unican.es>

Samuel Bernardo <samuel@lip.pt>

Foreword on fostering SQA practices in EOOSC



The EOOSC-Synergy project..

1. Delivers **2 baselines of QA criteria**

- Software <https://github.com/indigo-dc/sqa-baseline>
- Services <https://github.com/EOOSC-synergy/service-qa-baseline>

NEW

2. Develops an automated solution for assessing QA

→ **SQA as a Service** (aka SQAaaS)

- Development → github.com/eosc-synergy
- Based on CI/CD practices (DevOps)
- Aligned with the former SW & SVC baselines

More on this in tomorrow's talk "A quality based approach to software and services"
(EGI Core Services Roadmap - Part 2 session, 16:15 CET)

JePL (Jenkins Pipeline Library)



What?

- Core engine of the prospective SQAaaS solution, uses Jenkins PaC
 - Can be used independently from SQAaaS, we will see how!
- Provides the means to implement the reqs from SW & SVC baselines
 - Audience → EOSC service developers, any computational scientist or RSE

Why?

- Facilitates the adoption of DevOps practices → Improve SW & SVC QA
 - Compliance with SW & SVC baselines require CI/CD pipelines!

How?

- Uses (human-readable) YAML format, instead of Jenkins PaC DSL
 - Still, a fixed `Jenkinsfile` is needed
- Uses Docker Compose to orchestrate required services

Where? → <https://github.com/indigo-dc/jenkins-pipeline-library>

What is this demo about?

Goal: by using JePL, run a pipeline that checks the compliance of 3 types of criteria from the SW baseline

[qc_style, qc_coverage, qc_security]

- We will mimic the process of JePL adoption by a first-timer
 - Following the step-by-step guide at:
 - <https://indigo-dc.github.io/jenkins-pipeline-library/2.0.0/index.html>
 - Due to time constraints, the required files are already composed
- We will use a sample application delivered through the EOSC portal
 - DEEP as a Service: <https://github.com/indigo-dc/DEEPaaS>
- Results will appear in EOSC-Synergy's Jenkins instance
 - <https://jenkins.eosc-synergy.eu/job/eosc-synergy-org/job/DEEPaaS/>

Step-by-step guide



1. Let's start cloning the code repository (from fork's master branch):

```
git clone -b master git@github.com:EOOSC-synergy/DEEPaaS.git
```

2. Create an “jep1_demo” branch for the JePL-required files:

```
cd DEEPaaS && git checkout -b jep1_demo
```

3. Create [config.yml](#) and [docker-compose.yml](#) under the *.sqa* folder (pre-composed files, “eosc-synergy” branch):

```
mkdir .sqa && wget -P .sqa
```

```
https://raw.githubusercontent.com/EOOSC-synergy/DEEPaaS/eosc-synergy/.sqa/config.yml
```

```
https://raw.githubusercontent.com/EOOSC-synergy/DEEPaaS/eosc-synergy/.sqa/docker-compose.yml
```

4. Create the [Jenkinsfile](#) in the repo root path:

```
wget
```

```
https://raw.githubusercontent.com/EOOSC-synergy/DEEPaaS/eosc-synergy/Jenkinsfile
```

Step-by-step guide



5. We only need to commit and push the previous changes:

```
git add .sqa Jenkinsfile
```

```
git commit -m "Initial skeleton of JePL files"
```

```
git push -u origin jep1_demo
```

Now we can see the magic happening and wait for the results

https://jenkins.eosc-synergy.eu/job/eosc-synergy-org/job/DEEPaaS/job/jep1_demo/

While we wait for the results..



Some remarks..

- JePL provides **easy adoption of the QA criteria compiled in the SW and SVC baselines**
 - Hence, fostering SQA practices on research software, e.g. EOSC services
 - EOSC-Synergy Thematic Services are gradually adopting JePL
- JePL requires 3 files, but only one is the **fundamental basis**→**config.yml**
 - Jenkinsfile & docker-compose.yml are dependencies for automation & resource provisioning, respectively
- **Upcoming releases** of JePL will focus on supporting:
 - Additional QA criteria from the SW and SVC baselines
 - Additional composers for the CD part (k8s the first on the list)
- The **SQAaaS solution will leverage JePL to graphically compose on-demand CI/CD pipelines**
 - The JePL 3-file structure will be composed under the hood

Thanks for your attention