2023-11 - ONAP: Java 17 Upgrade for CPS and Policy Framework

Topic Leader(s)

- Gerard Nugent
- Adheli Tavares

Topic Description

30 min: Gerard Nugent

As part of Security and Dependencies upgrade deliveries, Policy Fmwk and CPS went through a Java version upgrade, from 11 to 17. In this presentation we bring the challenges, what went well, what could've improved and some snapshots of the applications, on how the new Java version affected their performance.

Topic Overview

We go through the motivation for the upgrade, the challenges in matching dependencies (Java version x Spring version x Jakarta EE changes), what tools we used to make the upgrade easier, what went well, what could've done better.

- Motivation:
    - Security reasons: lots of dependencies that were with vulnerabilities issues had the newer version developed with Java 17 as minimum version, like Spring Fmwk.
    - Jakarta EE upgrade from Javax
    - Deprecation on Junit4
    - Improve benchmarks for application usage
- Challenges:
    - main challenge was to match servlet dependencies with the Jakarta EE version of things. Most dependencies created a new groupId or artifactId for retro compatibility.
    - dependencies that hadn't changed yet from javax.* to jakarta.*
    - too many options, but very few documentation from OpenAPI/Swagger on the upgrade. (just a flag useJakarta that wasn't in any document)
    - override dependencies from oparent
    - java17 changes on reflection and closed access to java basic packages caused some issues with unit tests and clm jobs
- Tools:
    - intellij was very useful with the function to migrate junit4 to junit5. paid version also can help with jakartaEE migration
    - nexusIQ plugin helped when testing which next version would be the best (less breaking code changes and less/none vulnerabilities)

Slides & Recording

- DTF2023.odp