TECOR: Automate the Testing of One Product on Many Configurations

Anatoly Vasilevskiy, Hui Song, Brice Morin
SINTEF Digital, Oslo, Norway
Email: first.last@sintef.no

I. Overview

Automated testing is a crucial aspect of DevOps. It allows continuously monitoring and improving the quality of a shipped product. Even though test cases have a high degree of coverage, they tend to test one single configuration of the product, in the same controlled environment. This rather artificial setup is neither representative of the many different configurations that exist for a given product, nor of the many environments where different instances of the product will be deployed. This makes it difficult to unveil interesting properties of the product *e.g.*, how does the system perform when configured differently, or how does it perform in a more constrained environment?

This demonstration presents our Test Configuration Executor (TECOR), a tool to automatically execute existing test cases under multiple environments.
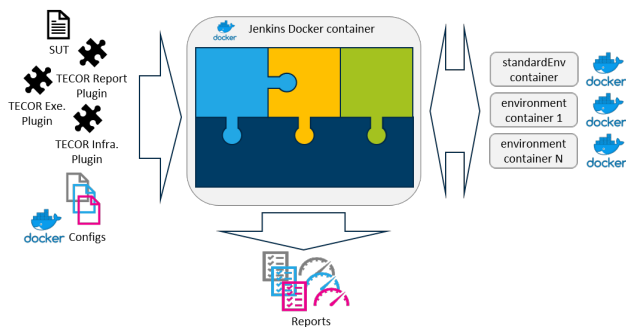

Fig. 1. TECOR - Conceptual architecture.

II. The Demonstration

The demonstration will show how to use TECOR to test XWiki, a web-based content management system implemented in Java. XWiki can be configured in multiple ways: different implementations and versions of the application server (such as Tomcat or Jetty), or of the database (such as MySQL, PostgreSQL or HSQL) can be combined. The objective of TECOR is to ensure the product can indeed support any combination and be as flexible as the developers claim.

Figure 1 shows the conceptual architecture of TECOR. The engineer provides a set of Docker files describing the different configurations to test against, building on top of the official Docker images of the platforms, in different versions.For each configuration, TECOR packages the software product together with the specified environment into a Docker image, and augments this environment with predefined instructions to build and test the product. Then, TECOR spawns a container for each image, builds and executes the tests on this container, and returns results. Finally, TECOR consolidates all results in a single report which the engineer can review and analyze.

The TECOR tool is freely available on GitHub[1] under the MIT license. We also submitted a screen-cast video[2] to show how we will demonstrate the testing of XWiki, together with another use case on a different software.

III. The Tool

TECOR is a light-weight tool promoting extensibility and re-usability, through a pluggable infrastructure, as shown in Figure 1. TECOR distinguishes the following plug-ins:

a) *execution plug-in*, which builds the SUT and executes test cases using a build system, e.g. maven,
b) *report plug-in*, which parses outputs of the execution plug-in and builds test reports,
c) *infrastructure plug-in*, which can set up a required infrastructure to perform testing properly.

Developers can implement their own plug-ins in the aforementioned three types, for their own systems. In addition, we ship TECOR Maven plug-ins which could be already used to test many projects.

TECOR can be integrated into existing CI/CD pipeline or run independently. In our prototype implementation, we have for example used a Jenkins pipeline. Jenkins executes the pipeline, which includes downloading TECOR, downloading a product to test, executing TECOR and publishing the execution reports. Setting up TECOR only requires an engineer to modify a few parameters in a configuration file, *i.e.* locations of a product to test, location of an execution plug-in, and location of the docker files. In addition, the engineer needs to set up Jenkins, or configure an existing one, to run TECOR. We provide a Docker image to simplify this set up. The TECOR repository contains detailed instructions and guidelines to set up TECOR together with Jenkins.

Acknowledgment

---

[1]https://github.com/STAMP-project/tecor
[2]https://www.youtube.com/watch?v=pZC4FMyidns