arm

TF-M
OpenCI

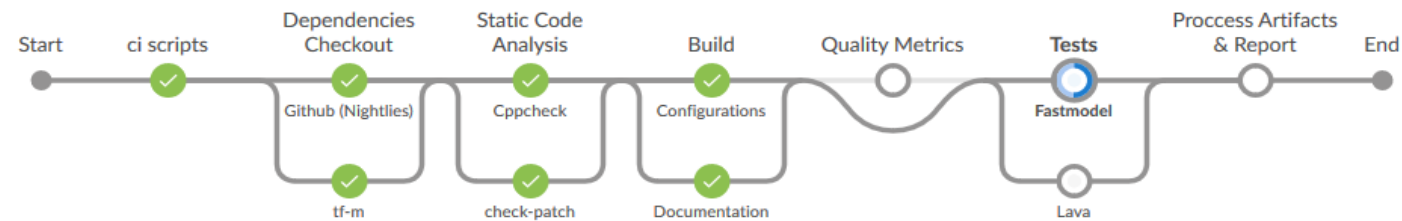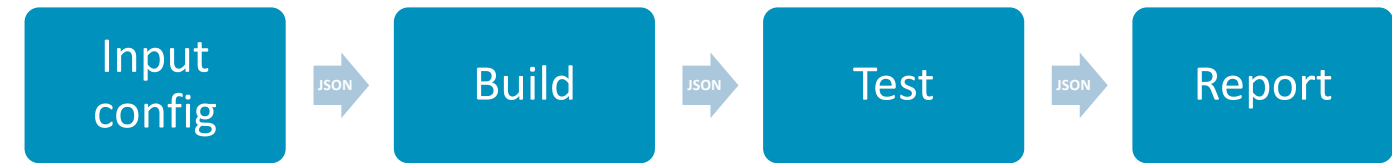A quick overview and future plans

Minos Galanakis
17/10/2019

CI Overview

Core features

- CI scripts designed to operate as a data flow.

- Jenkins unified pipeline, separates code and configuration.

- No information is discarded, every test is fully reproducible

| Input config | → JSON → | Build | → JSON → | Test | → JSON → | Report |

Start — ci scripts — Dependencies Checkout — Static Code Analysis — Build — Quality Metrics — Tests — Proccess Artifacts & Report — End

- Github (Nightlies)
- Cppcheck
- Configurations
- Fastmodel

- tf-m
- check-patch
- Documentation
- Lava

```
▼ build:
    ▼ AN521_GNUARM_ConfigCoreIPCTfmLevel2_Release_NOBL2:
        ▶ artefacts:                        {…}
        ▶ log:                              "/home/jenkins/workspace/…evel2_Release_NOBL2.log"
        ▶ build_cmd:                        "cmake -G \"Unix Makefile…uild ./ -- -j 2 install"
          status:                           "Success"
```

© 2019 Arm Limited

arm

Roadmap

Deployed features

- Building multiple configurations AN521, AN519, MUSCA_A1, MUCSA_B

- Testing on real hardware using the mainline golden configuration on MPS2 dev boards

- Building documentation (Sphynx and Doxygen)

- Different scope testing (per patch & nightly)

- Version control for dependencies

- Comprehensive report, captured in a key:value data structure, as well as extensive per build/test logs

Planned features

- Model testing

- User requested testing

- CI contribution process/ staging instance

- Extended platform support for real hardware testing

- More configurations/compilers

- Better coverage by static analysis tools

- Dashboard, capture and display of metrics about the build

arm

Discussion

Partner feedback is important

- Contribution strategies

- Script re-usability

- Offline testing

- Creation of job variants

- Scaling concerns

arm

arm

Thank You
Danke
Merci
谢谢
ありがとう
Gracias
Kiitos
감사합니다
धन्यवाद
شكرًا
תודה