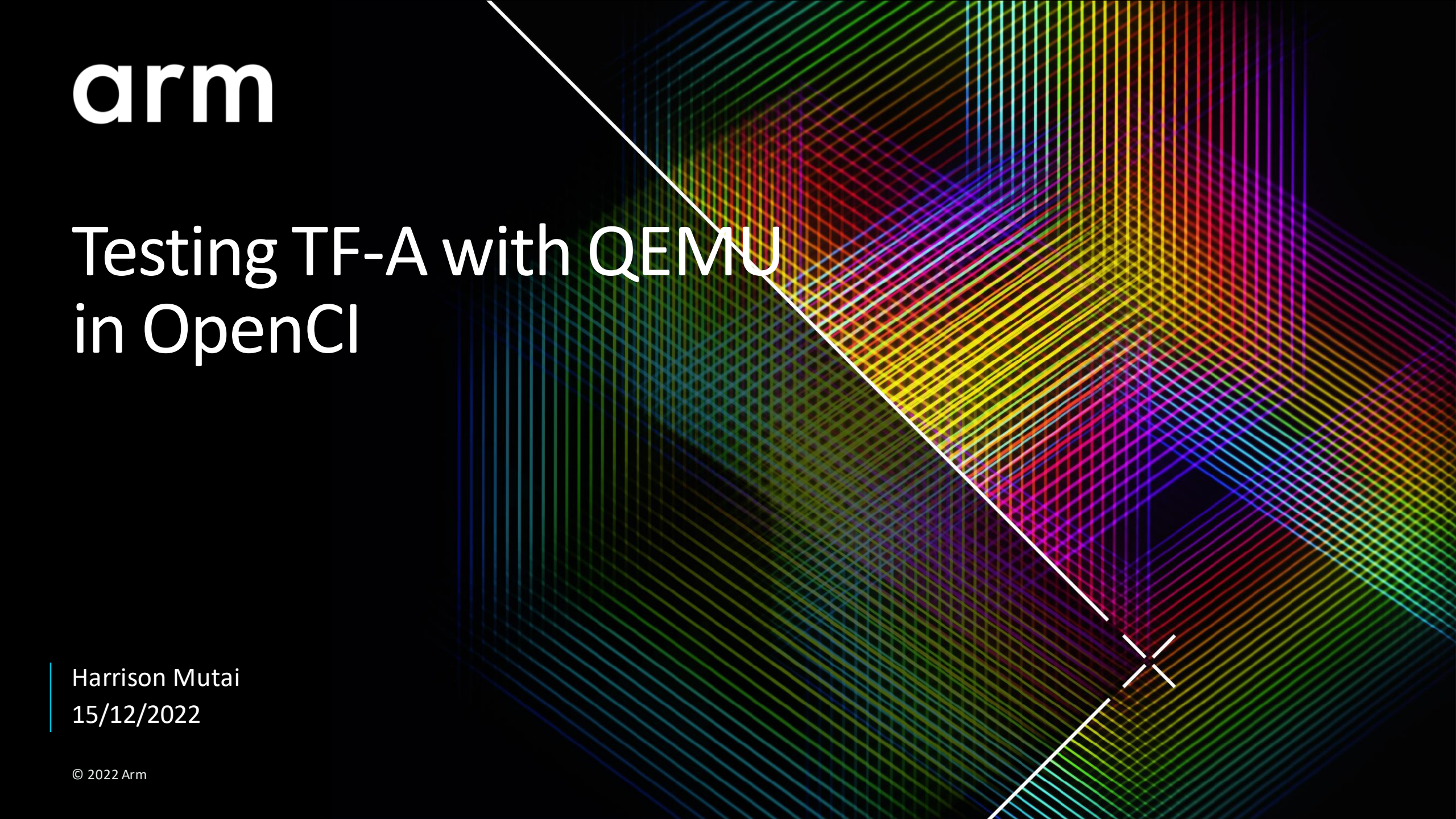


Testing TF-A with QEMU in OpenCI

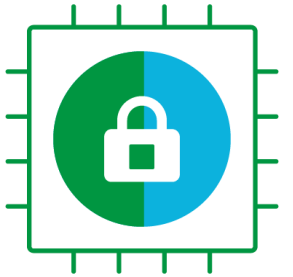
Harrison Mutai

15/12/2022

© 2022 Arm

Contents

- Running QEMU tests in OpenCI
  - Navigating the Jenkins UI
  - Navigating the test results
- Running local QEMU tests
- Changes in TF-A CI Scripts
  - Overview of the CI architecture
  - Changes to build and run packages
- Final Comments and Conclusions



© 2022 Arm

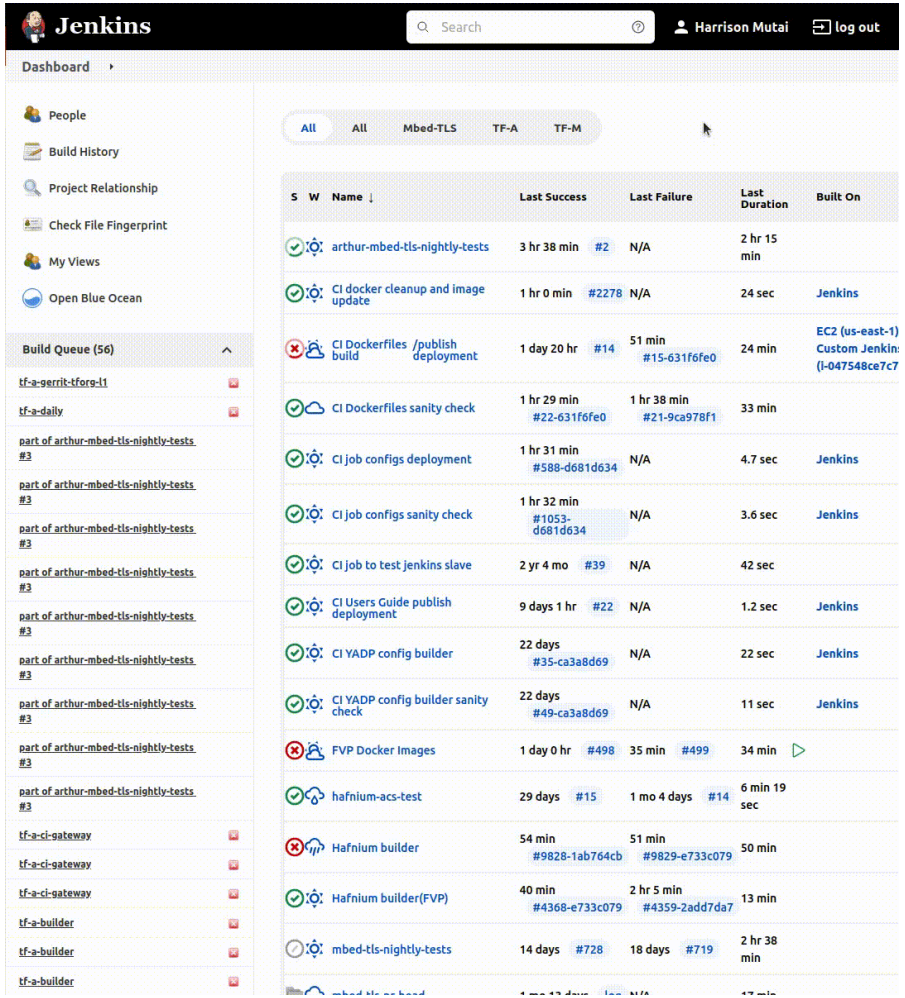
Running QEMU tests in OpenCI

© 2022 Arm

Running QEMU tests in OpenCI

Navigating the Jenkins UI

- Open the tf-a-ci-gateway job console in Jenkins: https://ci.trustedfirmware.org/job/tf-a-ci-gateway
- Select *Build with Parameters* from the left menu
- Enter **qemu-boot-tests** into the **TEST_GROUPS** input
- Hit build!

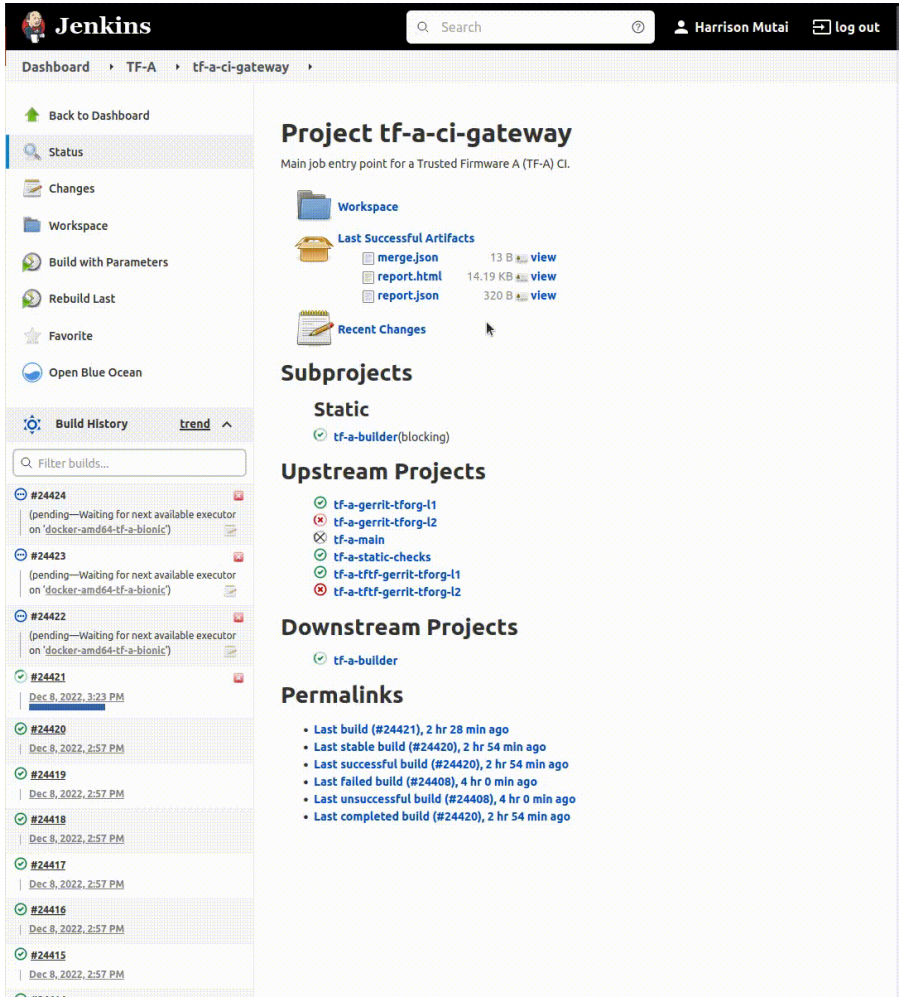


© 2022 Arm

Running QEMU tests in OpenCI

Navigating the Jenkins UI

- Open the tf-a-ci-gateway job console in Jenkins: https://ci.trustedfirmware.org/job/tf-a-ci-gateway
- Select *Build with Parameters* from the left menu
- Enter **qemu-boot-tests** into the **TEST_GROUPS** input
- Hit build!

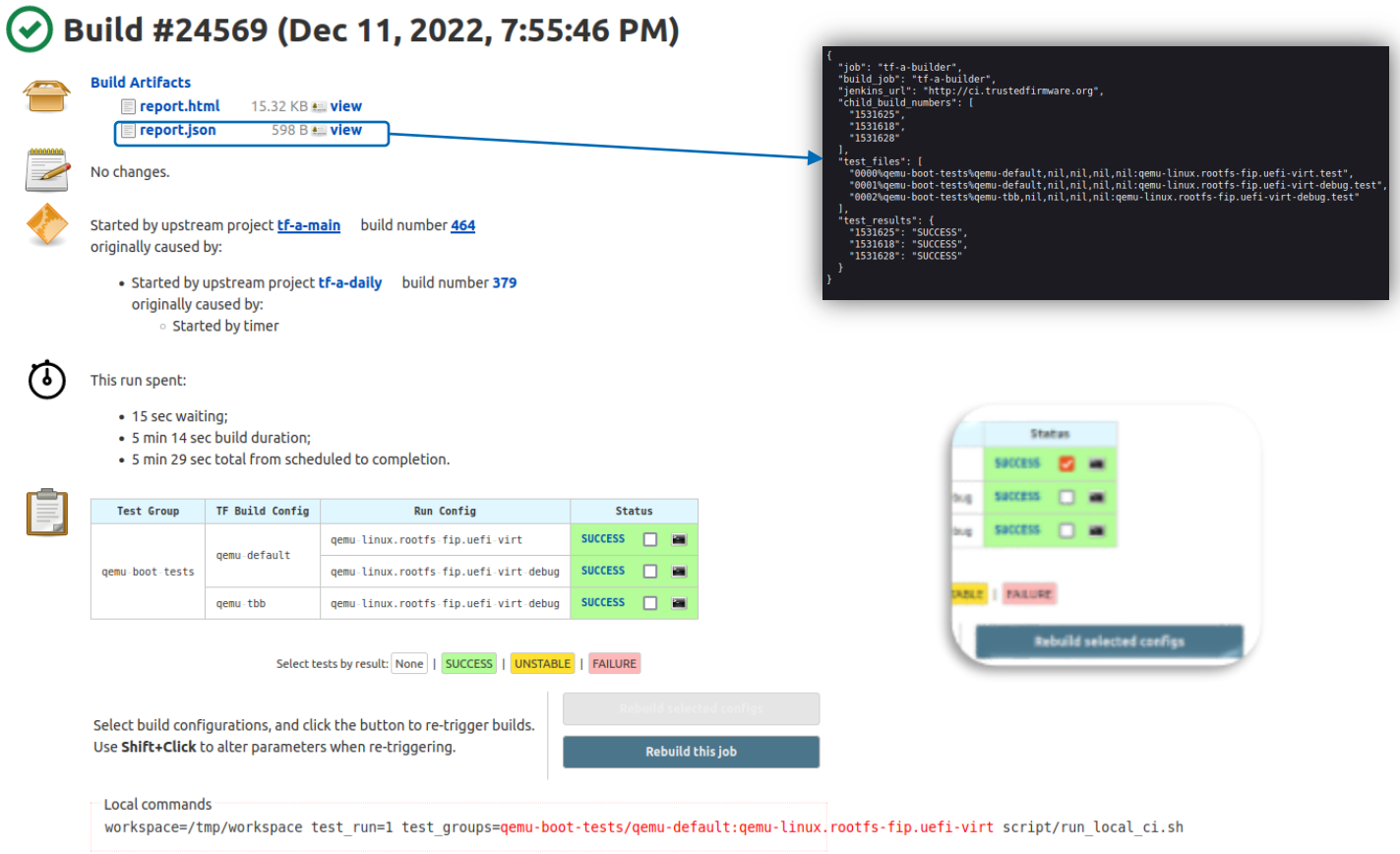


© 2022 Arm

Running QEMU tests in OpenCI

Navigating the test results

- An interactive GUI is produced by the `tf-a-ci-gateway` job
- Provides summary of individual test results
- Results also provided in a JSON-format file
- Specific test cases can be re-run directly from this console

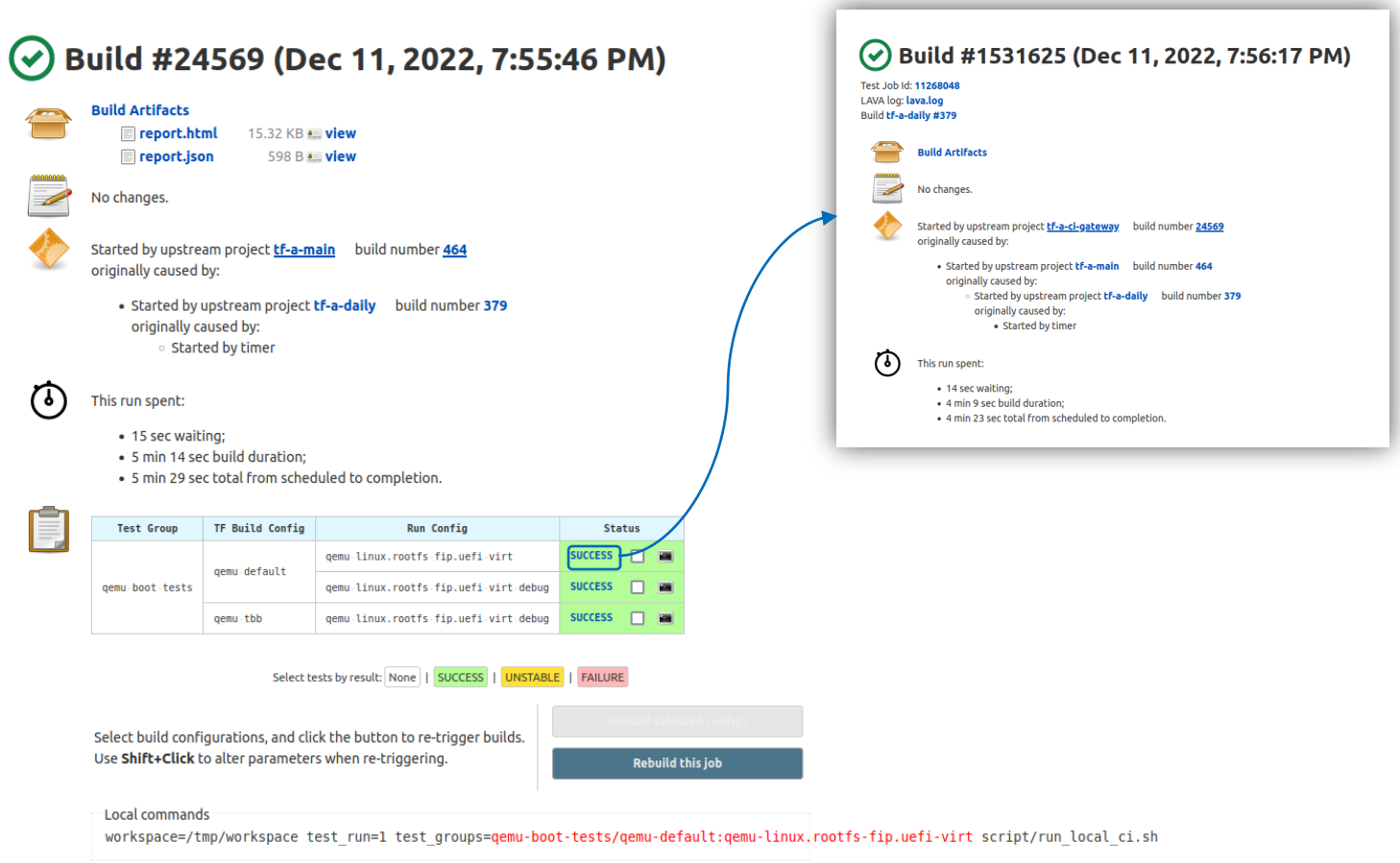


© 2022 Arm

Running QEMU tests in OpenCI

Navigating the test results

- The result in the status column is a link to the `tf-a-builder` that run the test
- Provides access to artefacts from the test and all logs
- Terminal symbol links to the console log of that job
- Results are retained for up to 15 days

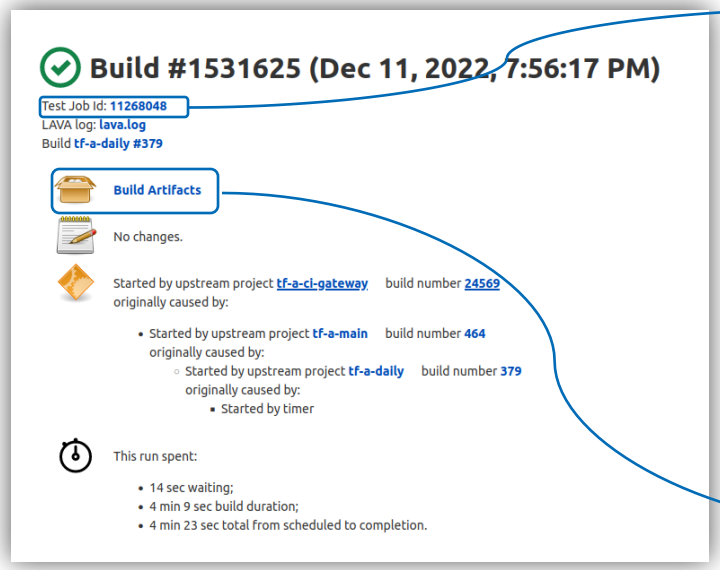


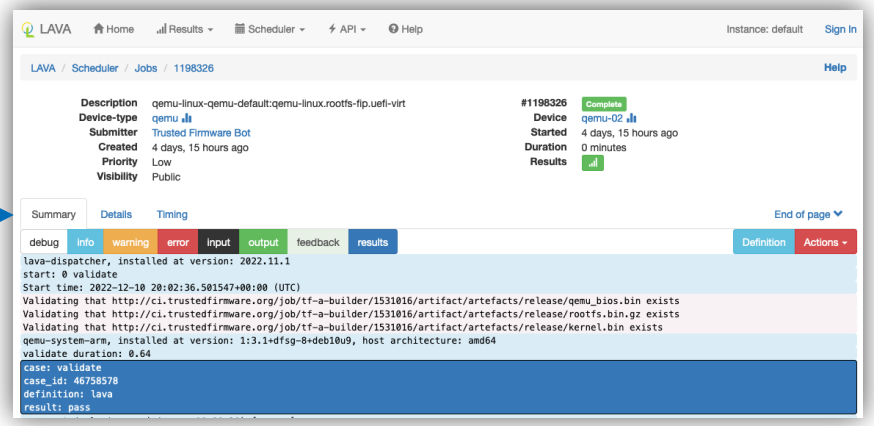
© 2022 Arm

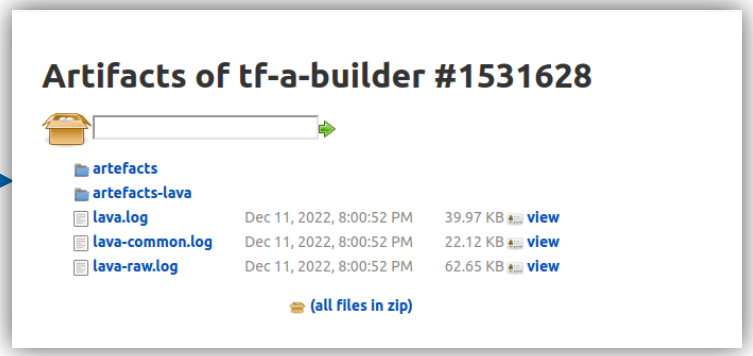
Running QEMU tests in OpenCI

Navigating the test results

- LAVA populates results live; the plain logs can be downloaded after the run.
- The interactive LAVA console is available as soon as the tf-a-builder submits the job to LAVA



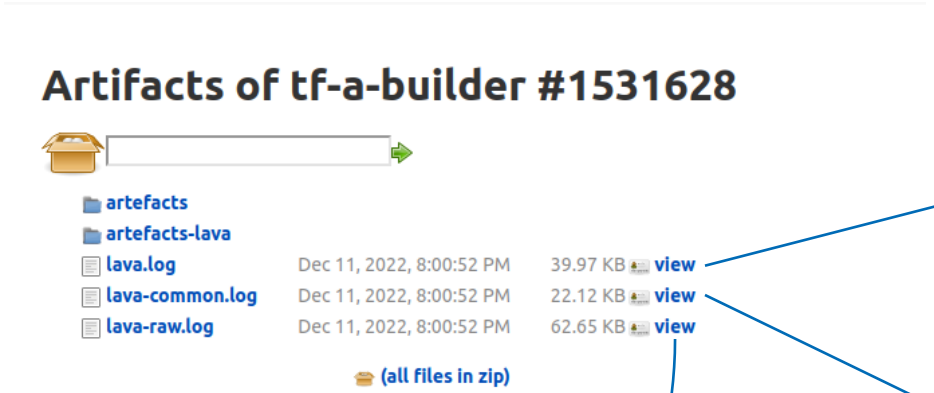




© 2022 Arm

Running QEMU tests in OpenCI

Navigating the test results




```
2022-12-11T19:59:42 lava-dispatcher, installed at version: 2022.11.1
2022-12-11T19:59:42 start: 0 validate
2022-12-11T19:59:42 Start time: 2022-12-11 19:59:42.927638+00:00 (UTC)
2022-12-11T19:59:42 Validating that http://ci.trustedfirmware.org/job/tf-a-builder/1531628/artifact/artefacts/debug/qemu_bios.bin exists
2022-12-11T19:59:43 Validating that http://ci.trustedfirmware.org/job/tf-a-builder/1531628/artifact/artefacts/debug/rootfs.bin.gz exists
2022-12-11T19:59:43 Validating that http://ci.trustedfirmware.org/job/tf-a-builder/1531628/artifact/artefacts/debug/kernel.bin exists
2022-12-11T19:59:43 qemu-system-arm, installed at version: 1:3.1+dfsg-8+deb10u9, host architecture: amd64
2022-12-11T19:59:43 validate duration: 0.58
2022-12-11T19:59:43 {'case': 'validate', 'definition': 'lava', 'result': 'pass'}
2022-12-11T19:59:43 start: 1 deployimages (timeout 00:00:30) [common]
2022-12-11T19:59:43 start: 1.1 download-retry (timeout 00:00:30) [common]
2022-12-11T19:59:43 start: 1.1.1 http-download (timeout 00:00:30) [common]
2022-12-11T19:59:43 downloading http://ci.trustedfirmware.org/job/tf-a-builder/1531628/artifact/artefacts/debug/qemu_bios.bin
2022-12-11T19:59:43 saving as /var/lib/lava/dispatcher/tmp/1201023/deployimages-uazuur2d/bios/qemu_bios.bin
2022-12-11T19:59:43 total size: 2495245 (2MB)
2022-12-11T19:59:43 No compression specified
2022-12-11T19:59:43 progress   1% (0MB)
```

```
NOTICE:  Booting Trusted Firmware
NOTICE:  BL1: v2.8(debug):v2.8-82-g25eb64726
NOTICE:  BL1: Built : 19:58:01, Dec 11 2022
INFO:    BL1: RAM 0xe0ee000 - 0xe0f8000
WARNING: BL1: cortex_a57: CPU workaround for 813419 was missing!
WARNING: BL1: cortex_a57: CPU workaround for 817169 was missing!
INFO:    BL1: cortex_a57: CPU workaround for disable_ldnp_overread was applied
WARNING: BL1: cortex_a57: CPU workaround for 826974 was missing!
WARNING: BL1: cortex_a57: CPU workaround for 826977 was missing!
WARNING: BL1: cortex_a57: CPU workaround for 828024 was missing!
WARNING: BL1: cortex_a57: CPU workaround for 829520 was missing!
WARNING: BL1: cortex_a57: CPU workaround for 833471 was missing!
WARNING: BL1: cortex_a57: CPU workaround for 859972 was missing!
WARNING: BL1: cortex_a57: CPU workaround for 1319537 was missing!
INFO:    BL1: cortex_a57: CPU workaround for cve_2017_5715 was applied
INFO:    BL1: cortex_a57: CPU workaround for cve_2018_3639 was applied
INFO:    BL1: cortex_a57: CPU workaround for cve_2022_23960 was applied
INFO:    Using crypto library 'mbed TLS'
INFO:    BL1: Loading BL2
INFO:    Loading image id=6 at address 0xe06b000
```


```
- {"dt": "2022-12-11T19:59:42.927082", "lvl": "info", "msg": "lava-dispatcher, installed at version: 2022.11.1"}
- {"dt": "2022-12-11T19:59:42.927436", "lvl": "info", "msg": "start: 0 validate"}
- {"dt": "2022-12-11T19:59:42.927653", "lvl": "info", "msg": "Start time: 2022-12-11 19:59:42.927638+00:00 (UTC)"}
- {"dt": "2022-12-11T19:59:42.927891", "lvl": "debug", "msg": "Validating that http://ci.trustedfirmware.org/job/tf-a-builder/1531628/artifact/artefacts/debug/qemu_bios.bin exists"}
- {"dt": "2022-12-11T19:59:43.116222", "lvl": "debug", "msg": "Validating that http://ci.trustedfirmware.org/job/tf-a-builder/1531628/artifact/artefacts/debug/rootfs.bin.gz exists"}
- {"dt": "2022-12-11T19:59:43.283045", "lvl": "debug", "msg": "Validating that http://ci.trustedfirmware.org/job/tf-a-builder/1531628/artifact/artefacts/debug/kernel.bin exists"}
- {"dt": "2022-12-11T19:59:43.504057", "lvl": "info", "msg": "qemu-system-arm, installed at version: 1:3.1+dfsg-8+deb10u9, host architecture: amd64"}
- {"dt": "2022-12-11T19:59:43.506511", "lvl": "info", "msg": "validate duration: 0.58"}
- {"dt": "2022-12-11T19:59:43.507002", "lvl": "results", "msg": {"case": "validate", "definition": "lava", "result": "pass"}}
- {"dt": "2022-12-11T19:59:43.507835", "lvl": "info", "msg": "start: 1 deployimages (timeout 00:00:30) [common]"}
- {"dt": "2022-12-11T19:59:43.508349", "lvl": "debug", "msg": "start: 1.1 download-retry (timeout 00:00:30) [common]"}
- {"dt": "2022-12-11T19:59:43.508841", "lvl": "debug", "msg": "start: 1.1.1 http-download (timeout 00:00:30) [common]"}
- {"dt": "2022-12-11T19:59:43.509546", "lvl": "info", "msg": "downloading http://ci.trustedfirmware.org/job/tf-a-builder/1531628/artifact/artefacts/debug/qemu_bios.bin"}
- {"dt": "2022-12-11T19:59:43.509970", "lvl": "debug", "msg": "saving as /var/lib/lava/dispatcher/tmp/1201023/deployimages-uazuur2d/bios/qemu_bios.bin"}
- {"dt": "2022-12-11T19:59:43.510327", "lvl": "debug", "msg": "total size: 2495245 (2MB)"}
- {"dt": "2022-12-11T19:59:43.510461", "lvl": "debug", "msg": "No compression specified"}
- {"dt": "2022-12-11T19:59:43.690483", "lvl": "debug", "msg": "progress   1% (0MB)"}
- {"dt": "2022-12-11T19:59:43.717838", "lvl": "debug", "msg": "progress   6% (0MB)"}
```


© 2022 Arm

Running local QEMU tests

© 2022 Arm

Running local QEMU tests

- Trust Firmware A and CI repository sources (v2.7 onwards).
- For existing tests, TFTF isn't strictly necessary (pass dummy value).
- System QEMU for Aarch64 (> v4.2.1).

```
test_run=1 \
workspace=$(mktemp -d) \
tfa_downloads="https://downloads.trustedfirmware.org/tf-a" \
nfs=$workspace \
tf_root="/path/to/trusted-firmware-a/" \
tftf_root="/path/to/tf-a-tests/" \
test_groups=qemu-boot-tests/qemu-default:qemu-linux.rootfs-fip.uefi-virt \
    bash $ci_root/script/run_local_ci.sh
```


© 2022 Arm

Running local QEMU tests

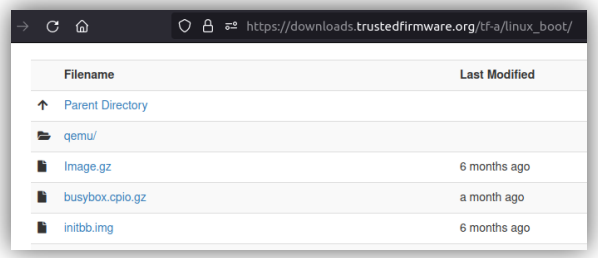
```
test_run=1 \
workspace=$(mktemp -d) \
tfa_downloads="https://downloads.trustedfirmware.org/tf-a" \
nfs=$workspace \
tf_root="/path/to/trusted-firmware-a/" \
tftf_root="/path/to/tf-a-tests/" \
test_groups=qemu-boot-tests/qemu-default:qemu-linux.rootfs-fip.uefi-virt \
    bash $ci_root/script/run_local_ci.sh
```

Do not spawn expect instances to track serial output!

Directory to build and run test from.

OpenCI parameter - scratch & tool area

Individual test we would like to run.



© 2022 Arm

Running local QEMU tests

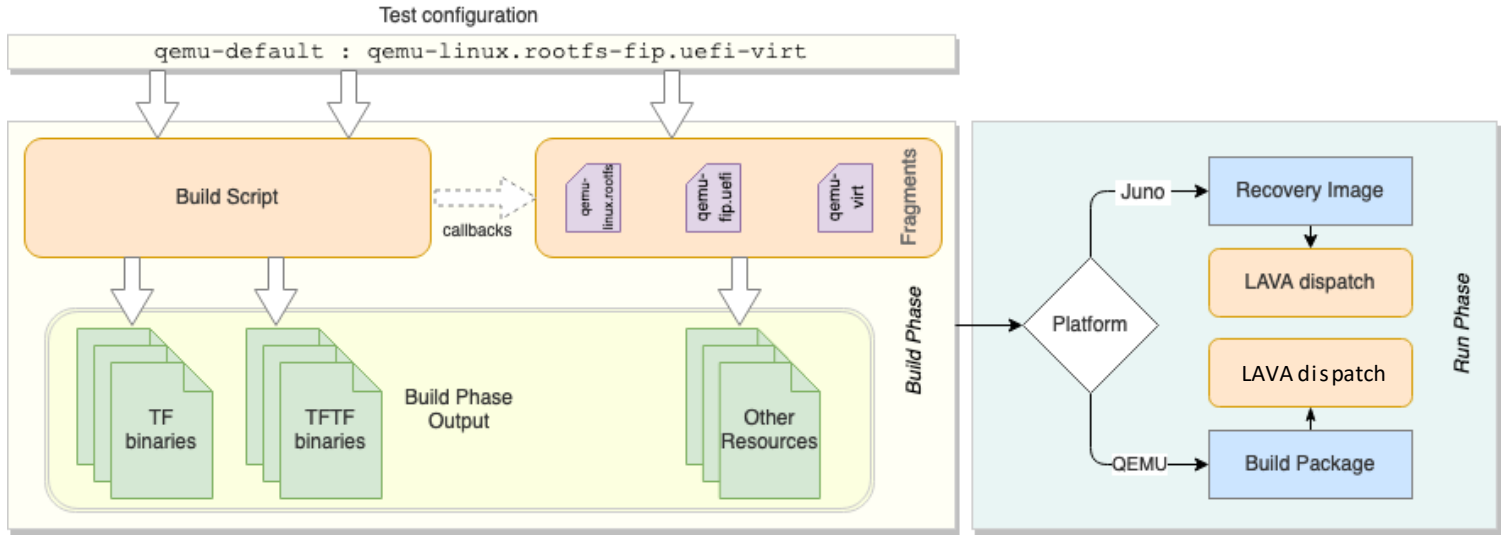
© 2022 Arm

Changes in TF-A CI Scripts

© 2022 Arm

Changes in TF-A CI Scripts

Overview of the CI architecture



- Consists of a build and run phase.
- Build phase downloads and generates all the resources required to run the test.
- The run phase executes the output of the build phase.
- Information about the build phase is encoded in the test configuration.

© 2022 Arm

Changes in TF-A CI Scripts

Overview of the CI architecture

```
1   CROSS_COMPILE=aarch64-none-elf-
2   PLAT=qemu
```

platform-ci/tf_config/qemu-default

{tf-build-config | nil}[,tftf-build-config]:{run-config | nil}

Build configuration – plain text files in `tf_config` or `tftf_config` containing build parameters for TF-A and or TFTF.

```
platform-ci/tf_config/
├── qemu-aarch32-clang-opteed
├── qemu-aarch32-opteed
├── qemu-default
├── qemu-gicv3
├── qemu-mb_hash256
├── qemu-opteed-tbb-enc
├── qemu_sbsa-default
├── qemu_sbsa-stack-protector
├── qemu_sbsa-stack-protector-rndcanary
└── qemu-tbb
```

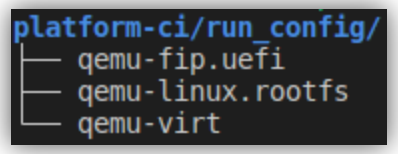

© 2022 Arm
arm

Changes in TF-A CI Scripts

Overview of the CI architecture

{tf-build-config | nil}[,tftf-build-config]:{run-config | nil}

Run configuration – specifies the steps necessary to setup the test environment i.e., choosing a non-secure payload.

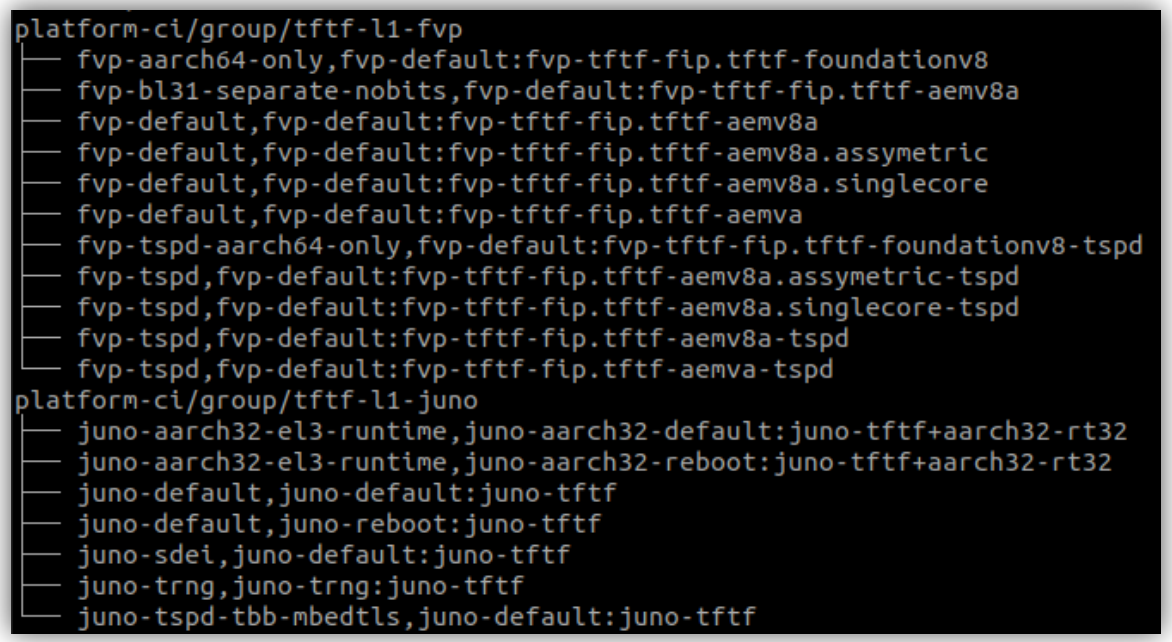


© 2022 Arm

Changes in TF-A CI Scripts

Overview of the CI architecture

- Test configurations of a similar nature are collected as a group.
- See the **group** subdirectory in the CI repository for already existing groups.
- Providing a group name to the CI scripts executes all tests thereunder i.e. qemu-boot-tests runs all QEMU tests under group/qemu-boot-tests



© 2022 Arm

Changes in TF-A CI Scripts

Changes to build and run packages

1. **Boot QEMU from TF-A to the Linux shell on a local machine**
2. Automate build steps and fetching of pre-built resources
3. Setup reproducible test environment in CI
4. Automate job submission

© 2022 Arm

Changes in TF-A CI Scripts

Changes to build and run packages

- There are two ways to launch QEMU guests: firmware based and kernel shortcut.
- **`-kernel`** in addition to **`-bios`** means that QEMU loads the kernel into DRAM and puts the kernel info into the **`fw_cfg`** device
- Bios is responsible for providing the kernel with the device tree
- A binary consisting of BL1 and the FIP are used as the BootROM. Bootloader built into the FIP image as BL33

```
$ qemu-system-aarch64 \
    -machine 'secure=on,virtualization=on,gic-version=2' -cpu max \
    -nographic -display none -d unimp \
    -m 4G -M virt -smp 4 \
    -append 'console=ttyAMA0,115200n8 root=/dev/vda earlycon' \
    -bios qemu_bios.bin -initrd rootfs.bin -kernel kernel.bin
```

© 2022 Arm
arm

Changes in TF-A CI Scripts

Changes to build and run packages

1. ~~Boot QEMU from TF-A to the Linux shell on a local machine~~
2. **Automate build steps and fetching of pre-built resources**
3. Setup reproducible test environment in CI
4. Automate job submission

© 2022 Arm

Changes in TF-A CI Scripts

Changes to build and run packages

- What parts do we need?
  - Kernel
  - EDK II
  - Root filesystem
  - Boot image (bl1.bin + fip.bin)
- Some parts not possible to build at runtime ie kernel and EDK II
- Must build TF-A from source!

```
$ qemu-system-aarch64 \
    -machine 'secure=on,virtualization=on,gic-version=2' -cpu max \
    -nographic -display none -d unimp \
    -m 4G -M virt -smp 4 \
    -append 'console=ttyAMA0,115200n8 root=/dev/vda earlycon' \
    -bios qemu_bios.bin -initrd rootfs.bin -kernel kernel.bin
```

© 2022 Arm
arm

Changes in TF-A CI Scripts

Changes to build and run packages

- Introduced three fragments:
  - `run_config/qemu-linux.rootfs`
  - `run_config/qemu-fip.uefi`
  - `run_config/qemu-virt`
- Fetch and archive kernel, and root filesystem
- Fetch EDK II and build the FIP with it as BL33
- Generate the BootROM that is provided with the **`-bios`** option

```
14  post_tf_archive(){
15      image="qemu_bios.bin" bl1_path="$archive/bl1.bin" \
16          fip_path="$archive/fip.bin" gen_qemu_image
17  }
```

`run_config/qemu_fip.uefi`

```
72  gen_qemu_image(){
73      local image=${image:?}
74      local bl1_path=${bl1_path:?}
75      local fip_path=${fip_path:?}
76
77      # Cocatenate bl1 and fip images to create a single BIOS consumed by QEMU.
78      cp $bl1_path "$image"
79      dd if=$fip_path of="$image" bs=64k seek=4
80
81      archive_file "$image"
82  }
```

`qemu_utils.sh`

© 2022 Arm

Changes in TF-A CI Scripts

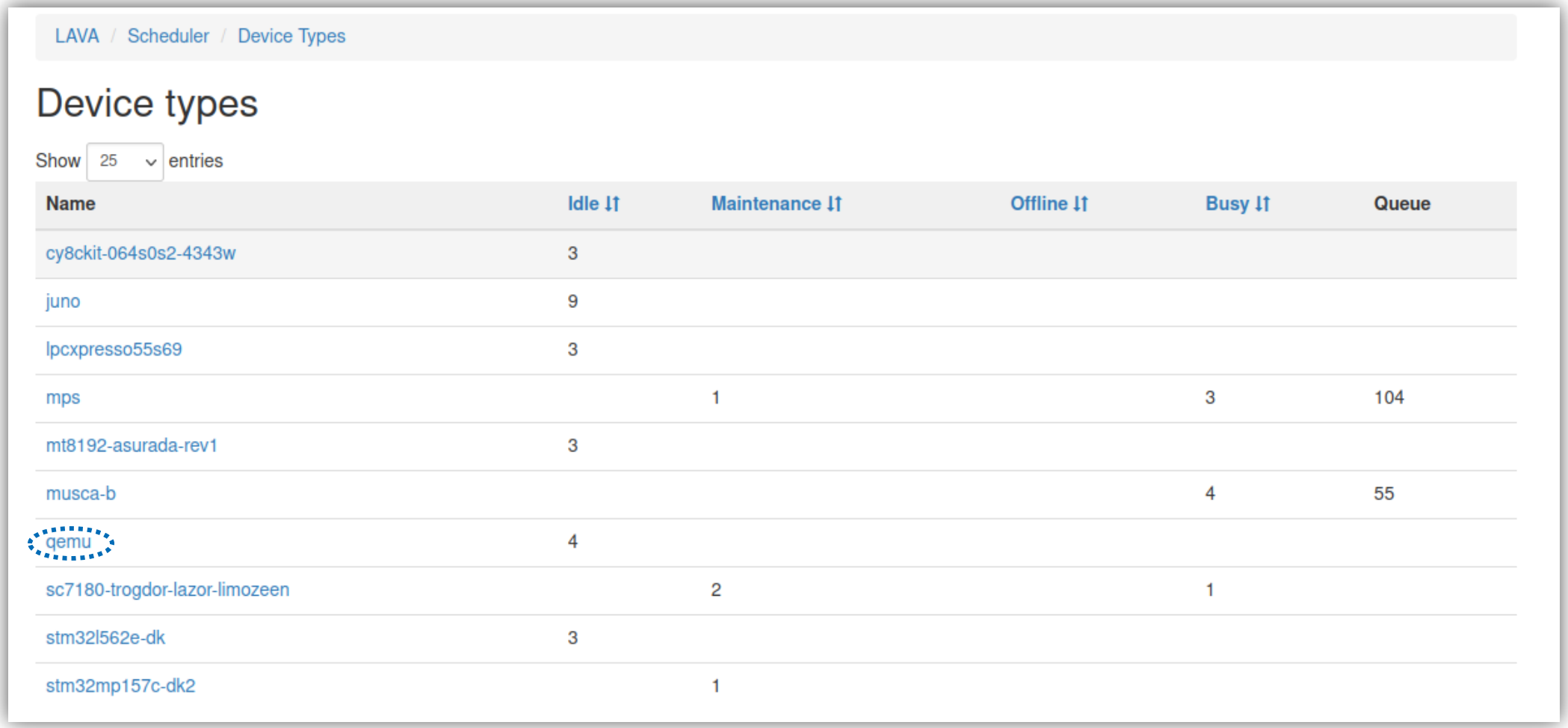
Changes to build and run packages

1. ~~Boot QEMU from TF-A to the Linux shell on a local machine~~
2. ~~Automate build steps and fetching of pre-built resources~~
3. **Setup reproducible test environment in CI**
4. Automate job submission

© 2022 Arm

Changes in TF-A CI Scripts

Changes to build and run packages



LAVA / Scheduler / Device Types

Device types

Show 25 entries

Name	Idle	Maintenance	Offline	Busy	Queue
cy8ckit-064s0s2-4343w	3				
juno	9				
lpcxpresso55s69	3				
mps		1		3	104
mt8192-asurada-rev1	3				
musca-b				4	55
qemu	4				
sc7180-trogdor-lazor-limozeen		2		1	
stm32l562e-dk	3				
stm32mp157c-dk2		1			

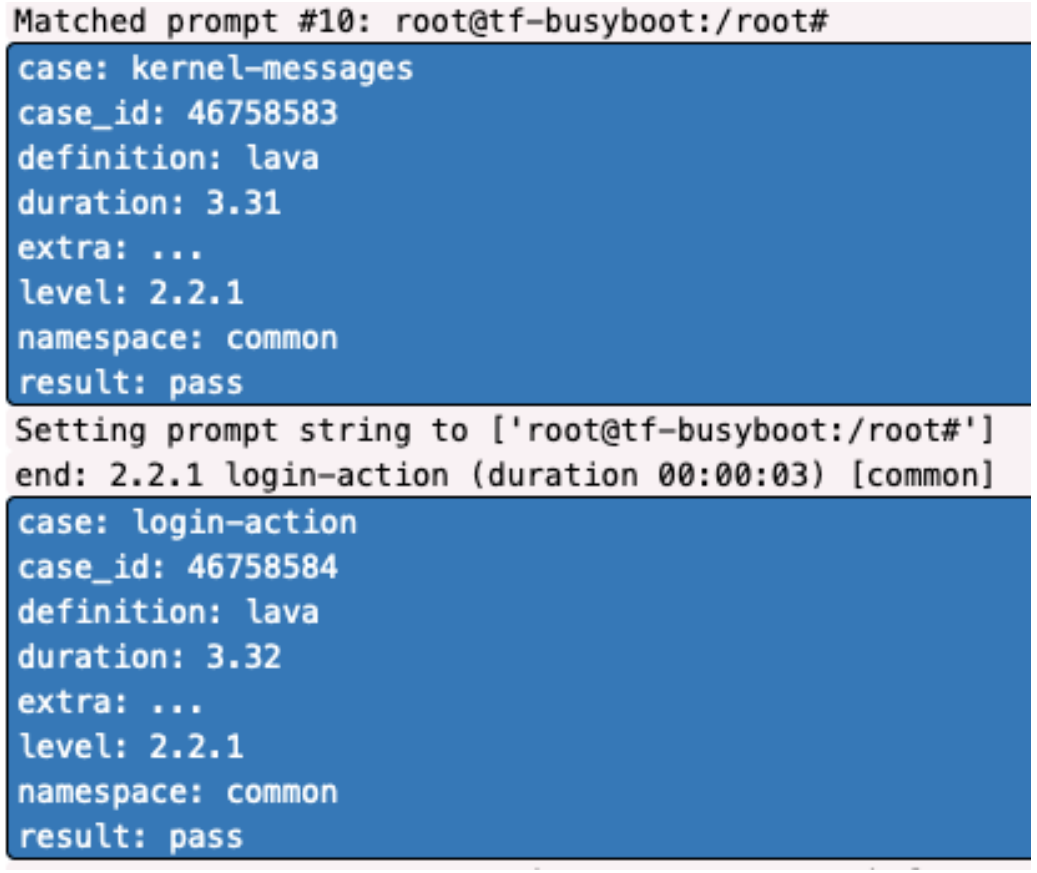
https://tf.validation.linaro.org/scheduler/device_types

© 2022 Arm

Changes in TF-A CI Scripts

Changes to build and run packages

- Each test job has a list of actions [1]:
  - Boot – boot the device and check prompts to see whether device has booted correctly
  - Deploy – download files to boot the device and prepare an overlay of files to run the test action
  - Test – clone and run tests for this job
- For boot action, it constructs the correct pipeline from a specified set
- User provides prompt to expect when system boots successfully… LAVA fails if it doesn't find this.



Boot action prompt check in LAVA UI

© 2022 Arm

Changes in TF-A CI Scripts

Changes to build and run packages

1. ~~Boot QEMU from TF-A to the Linux shell on a local machine~~
2. ~~Automate build steps and fetching of pre-built resources~~
3. ~~Setup reproducible test environment in CI~~
4. **Automate job submission**

© 2022 Arm

Changes in TF-A CI Scripts

Changes to build and run packages

- Jobs may be submitted to LAVA in one of three ways:
  1. Using the lavacli program
  2. XML-RPC API [2]
  3. Its web UI [3].
- LAVA jobs are defined in YAML files
- A job defines what software should be deployed on the device under test (DUT) and what actions should be performed there.
- Created a template job configuration for boot tests.

```
device_type: qemu
job_name: qemu-linux-${test_config}
priority: medium
visibility: public

metadata:
  test_config: ${test_config}
  build_url: ${BUILD_URL}
  emulator: ${model}

$(if [ -n "${gerrit_url}" ]; then
  cat <<-YAML
  gerrit_url: "${gerrit_url}"
  YAML
fi)

context:
  arch: aarch64
  machine: ${machine:-virt}
  cpu: ${cpu:-max}
  extra_options:
$(for boot_argument in "${boot_arguments[@]:?}"; do
  cat <<-YAML
      - ${boot_argument}
  YAML
done)
```

script/lava-templates/qemu-linux.yaml

© 2022 Arm

Changes in TF-A CI Scripts

Changes to build and run packages

```
device_type: qemu
job_name: qemu-linux-${test_config}
priority: medium
visibility: public

metadata:
  test_config: ${test_config}
  build_url: ${BUILD_URL}
  emulator: ${model}

$(if [ -n "${gerrit_url}" ]; then
  cat <<-YAML
  gerrit_url: "${gerrit_url}"
  YAML
fi)

context:
  arch: aarch64
  machine: ${machine:-virt}
  cpu: ${cpu:-max}
  extra_options:
$(for boot_argument in "${boot_arguments[@]:?}"; do
  cat <<-YAML
    - ${boot_argument}
  YAML
done)
```

```
generate_lava_job(){
    local model="qemu-virt"

    model="$model" gen_model_params
    model="$model" gen_qemu_yaml
}
```

```
gen_qemu_yaml(){
    model="${model:?}"
    model_bin="${model_bin:qemu-system-aarch64}"

    yaml_template_file="$workspace/qemu_template.yaml"
    yaml_file="$workspace/qemu.yaml"
    yaml_job_file="$workspace/job.yaml"
    lava_model_params="$workspace/lava_model_params"

    # this function expects a template, quit if it is not present
```

```
device_type: qemu
job_name: qemu-linux-qemu-default:qemu-linux.rootfs-fip.uefi-virt
priority: medium
visibility: public

metadata:
  test_config: qemu-default:qemu-linux.rootfs-fip.uefi-virt
  build_url:
  emulator: qemu-virt

context:
  arch: aarch64
  machine: virt
  cpu: max
  extra_options:
    - -machine 'secure=on,virtualization=on,gic-version=2'
    - -smp 4
    - -m 4G
    - -nographic -display none -d unimp
    - -append 'console=ttyAMA0,115200n8 root=/dev/vda earlycon'
```


© 2022 Arm

Conclusion

- To get started running tests, you might need to be added to this group:
  - https://github.com/orgs/trusted-firmware-ci/teams/trusted-firmware-a-openci-users/
  - Get in touch via the mailing list or directly to Joanna Farley to get added.
- Currently QEMU tests do not gate patch submission, long term strategy is to add them to L1/L2 tests once we are confident ther are stable
- There are still quite a few gaps that need to be filled
  - Expect script support
  - Measured Boot
  - OPTEE
  - U-Boot
- Interested, or keen to make it better? Contributions are welcome!
- We are also looking for some helping maintaining the infrastructure – if this is of interest, please do get in touch!

© 2022 Arm

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

© 2022 Arm

Resources

1. https://validation.linaro.org/static/docs/v2/explain_first_job.html
2. https://tf.validation.linaro.org/static/docs/v2/data-export.html#xml-rpc
3. https://tf.validation.linaro.org/static/docs/v2/first-job.html#web-ui-submit

© 2022 Arm

The Arm trademarks featured in this presentation are registered trademarks or trademarks of Arm Limited (or its subsidiaries) in the US and/or elsewhere. All rights reserved. All other marks featured may be trademarks of their respective owners.

www.arm.com/company/policies/trademarks

© 2022 Arm