tc testing with TuxSuite

October 30 2023,

Vishal Bhoj <vishal.bhoj@linaro.org>
Anders Roxell <anders.roxell@linaro.org>

What is TuxSuite™?

TuxSuite™ is a platform of easy to use, on-demand tools and APIs

- Provide Open Source tools
- Container based
- Building and testing the Linux Kernel, Android, OpenEmbedded and Mbed-TLS in parallel, at scale
- Produce portable and reproducible build(s) and/or test(s)
- Traditional approach: trade Off **throughput vs cost**
  -  throughput =>  builders =>  cost
  - cost =>  builders =>  throughput
  - Why?
    - **Most of the time, servers are idling but costly**
    - **Requires a lot of administration**
- tuxsuite approach: **fully dynamic system**

tuxsuite

Linaro

TuxSuite's Capabilities

Build

- Linux Kernel via TuxMake
- OpenEmbedded via TuxBake
- Android Kernel and AOSP via Google/Android build script
- Mbed-TLS via Mbed-TLS' build script

Test

- Run tests on virtual hardware via TuxRun (QEMU and FVP [Fixed Virtual Platforms])
- Run tests on real hardware via a LAVA lab as a backend (HW like juno, db845, rb5, x15, Rpi, E-850)
- Curated rootfs (debian, buildroot), test-suites (ltp, libgpiod, kvm-unit-tests,...) and QEMU master/releases

Plan file

- Concept of a build(s) triggers a test(s) via TuxSuite plan

Monitoring git tree's

- Config file that specifies what tree/branch to track and what plan file to trigger via TuxTrigger

Storage artifacts

- Storage the artifacts from the build and test, individual retention policy via TuxPub

Command line client

- To submit a build/test/plan via TuxSuite cli

TuxSuite Demo: Building

How to **build** the Linux kernel and kselftest from net-next, with clang-nightly easily

```
tuxsuite build
  --git-repo https://git.kernel.org/pub/scm/linux/kernel/git/netdev/net-next.git \
  --git-ref main --target-arch x86_64 \
  --toolchain clang-nightly LLVM=1 LLVM_IAS=1 \
  --kconfig defconfig \
  --kconfig
https://raw.githubusercontent.com/Linaro/meta-lkft/kirkstone/meta/recipes-kernel/linux/files/systemd.config \
  --kconfig tools/testing/selftests/tc-testing/config \
    debugkernel cpupower headers kernel kselftest modules
```

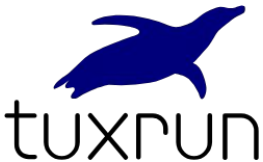
tuxmake

Linaro

TuxSuite Demo: Testing

How to test the built linux kernel and kselftest on QEMU.

```
tuxsuite test \
  --device qemu-x86_64 \
  --boot-args "'rw systemd.log_level=warning'" \
  --kernel
https://storage.tuxsuite.com/public/tuxsuite/vishal/builds/2XQzJrgiVJMce65rDS5VZAFKw5w/bzImage \
  --modules
https://storage.tuxsuite.com/public/tuxsuite/vishal/builds/2XQzJrgiVJMce65rDS5VZAFKw5w/modules.ta
r.xz \
  --rootfs https://storage.tuxboot.com/debian/bookworm/amd64/rootfs.ext4.xz \
  --parameters
KSELFTEST=https://storage.tuxsuite.com/public/tuxsuite/vishal/builds/2XQzJrgiVJMce65rDS5VZAFKw5w/
kselftest.tar.xz \
  --tests kselftest-tc-testing \
  --timeouts boot=15 --timeouts kselftest-tc-testing=20
```



Linaro

TuxSuite Demo: Plan

Tuxsuite build and test can be replaced with a plan file

```
tuxsuite plan \
  https://gitlab.com/vishalbhoj/tc-testing/-/raw/main/tc-x86.yaml \
  --git-repo https://git.kernel.org/pub/scm/linux/kernel/git/netdev/net-next.git \
  --git-ref main
```

... then we have to craft a plan file...

tc testing with TuxSuite - Plan

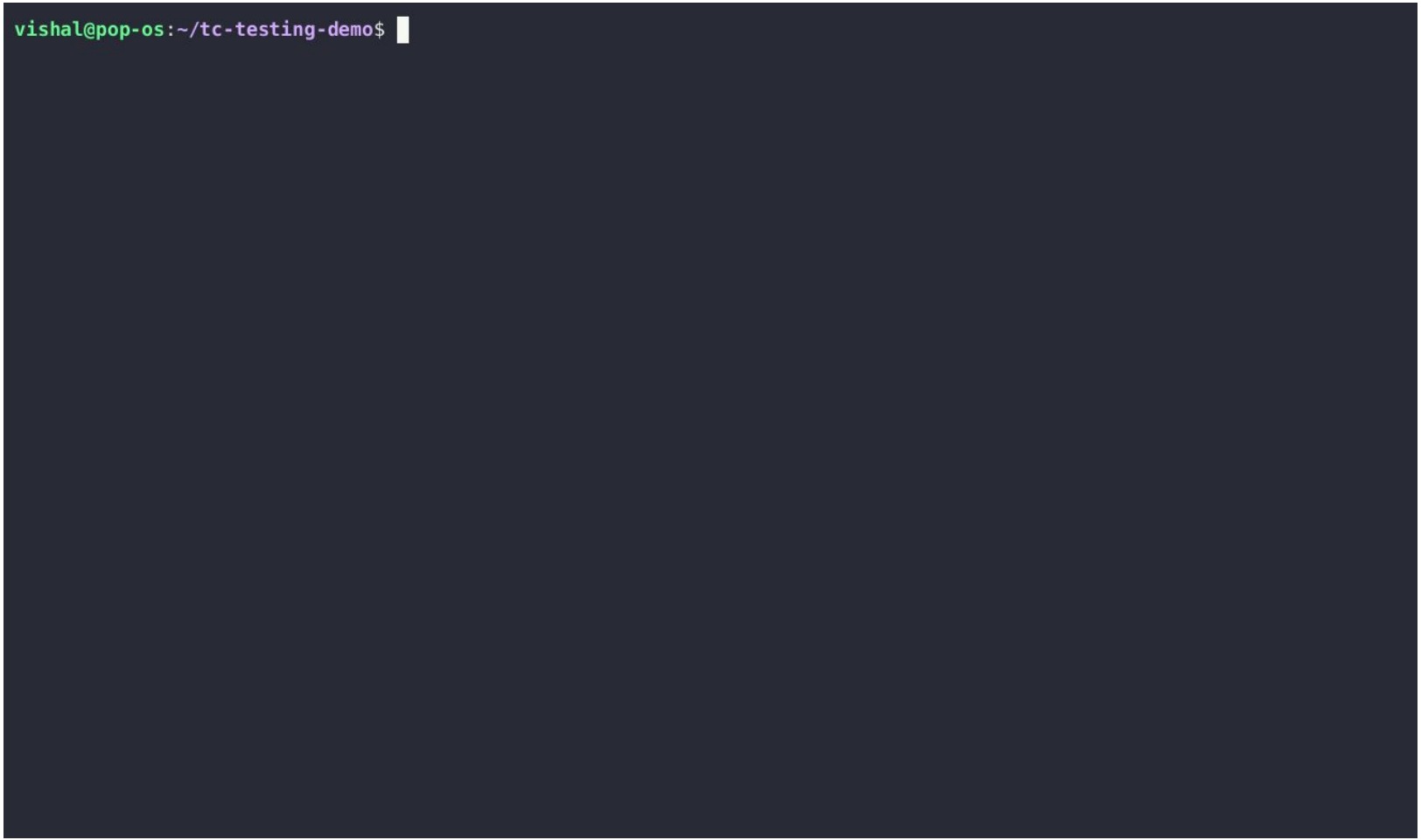
```
version: 1
name: Build kernel and kselftest.
description: Build and test linux kernel with clang-nightly run kselftest-tc-testing
jobs:
- name: x86-clang-lkftconfig-kselftest
  builds:
    - build_name: clang-nightly-lkftconfig-kselftest
      target_arch: x86_64
      toolchain: clang-nightly
      make_variables: {"LLVM": 1, "LLVM_IAS": 1}
      targets: ["debugkernel", "cpupower", "headers", "kernel", "kselftest", "modules"]
      kconfig: [ defconfig,
https://raw.githubusercontent.com/Linaro/meta-lkft/kirkstone/meta/recipes-kernel/linux/files/systemd.c
onfig, tools/testing/selftests/tc-testing/config ]
```

Linaro

tc testing with TuxSuite - Plan Cont...

```
version: 1
name: Build kernel and kselftest.
description: Build and test linux kernel with clang-nightly run kselftest-tc-testing
jobs:
- name: x86-clang-lkftconfig-kselftest
  builds:
    - build_name: clang-nightly-lkftconfig-kselftest
      target_arch: x86_64
      toolchain: clang-nightly
      make_variables: {"LLVM": 1, "LLVM_IAS": 1}
      targets: ["debugkernel", "cpupower", "headers", "kernel", "kselftest", "modules"]
      kconfig: [ defconfig,
https://raw.githubusercontent.com/Linaro/meta-lkft/kirkstone/meta/recipes-kernel/linux/files/systemd.c
onfig, tools/testing/selftests/tc-testing/config ]
  tests:
    - device: qemu-x86_64
      boot_args: "'rw systemd.log_level=warning'"
      parameters: {KSELFTEST: "$BUILD/kselftest.tar.xz"}
      rootfs: https://storage.tuxboot.com/debian/bookworm/amd64/rootfs.ext4.xz
      tests: [kselftest-tc-testing]
      timeouts: { "boot": 15, "kselftest-tc-testing": 20 }
```

Linaro


vishal@pop-os:~/tc-testing-demo$

Plan page

Parameters

name	Build kernel and kselftest.
description	Build and test linux kernel with clang-nightly run kselftest-tc-testing

Builds

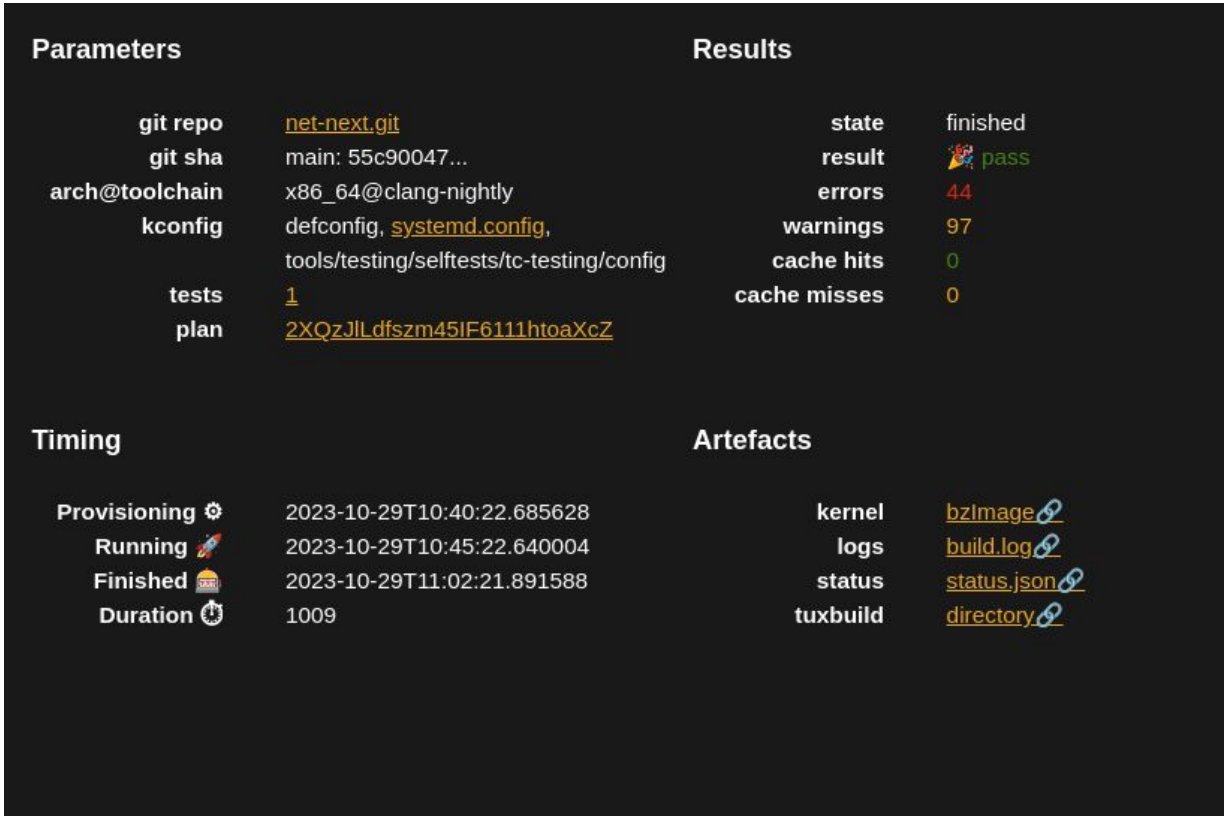
uid	name	toolchain	state	result
2XQzJrgiVJMce65rDS5VZAFKw5w	clang-nightly-l…	x86_64@clang-nightly	finished	🎉 pass

Tests

uid	device	tests	state	results
2XQzJyq4aeMzrEuGbtNaU1ZKh51	qemu-x86_64	kselftest-tc-testing	👹 fail	❌ kselftest-tc-testing

Linaro

Build page



Parameters

git repo	net-next.git
git sha	main: 55c90047...
arch@toolchain	x86_64@clang-nightly
kconfig	defconfig, systemd.config, tools/testing/selftests/tc-testing/config
tests	1
plan	2XQzJlLdfszm45IF6111htoaXcZ

Results

state	finished
result	🎉 pass
errors	44
warnings	97
cache hits	0
cache misses	0

Timing

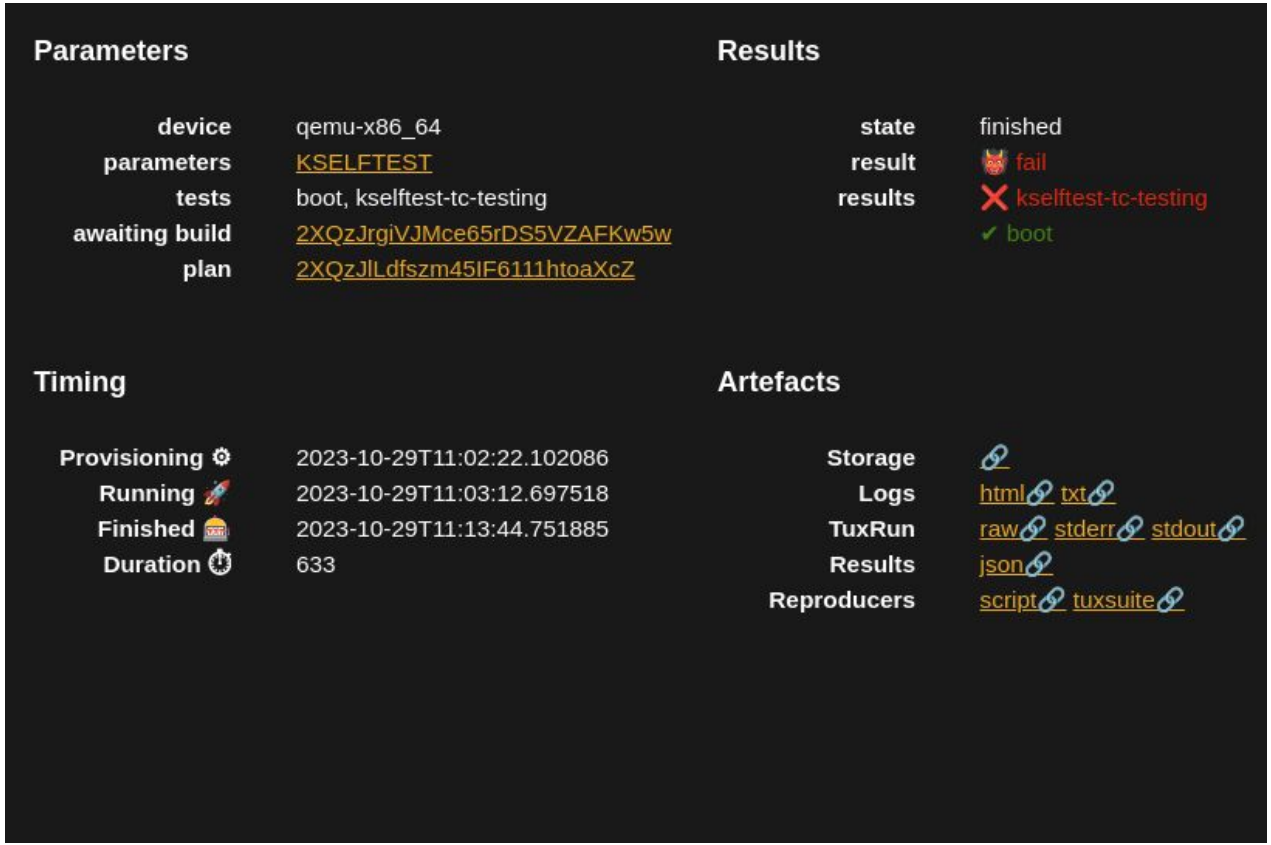
Provisioning ⚙	2023-10-29T10:40:22.685628
Running 🚀	2023-10-29T10:45:22.640004
Finished 🚋	2023-10-29T11:02:21.891588
Duration ⏱	1009

Artefacts

kernel	bzImage🔗
logs	build.log🔗
status	status.json🔗
tuxbuild	directory🔗

Linaro

Test page



Parameters

device	qemu-x86_64
parameters	KSELFTEST
tests	boot, kselftest-tc-testing
awaiting build	2XQzJrgiVJMce65rDS5VZAFKw5w
plan	2XQzJlLdfszm45IF6111htoaXcZ

Results

state	finished
result	👹 fail
results	❌ kselftest-tc-testing
	✔ boot

Timing

Provisioning ⚙	2023-10-29T11:02:22.102086
Running 🚀	2023-10-29T11:03:12.697518
Finished 🎰	2023-10-29T11:13:44.751885
Duration ⏱	633

Artefacts

Storage	🔗
Logs	html🔗 txt🔗
TuxRun	raw🔗 stderr🔗 stdout🔗
Results	json🔗
Reproducers	script🔗 tuxsuite🔗

Linaro

Local Build Reproducer

```
#!/bin/sh
# TuxMake is a command line tool and Python library that provides
# portable and repeatable Linux kernel builds across a variety of
# architectures, toolchains, kernel configurations, and make targets.
# TuxMake supports the concept of runtimes.
# See https://docs.tuxmake.org/runtimes/, for that to work it requires
# that you install podman or docker on your system.
# To install tuxmake to your home directory at ~/.local/bin:
# pip3 install -U --user tuxmake
# Or install a deb/rpm depending on the running distribution
# See https://tuxmake.org/install-deb/ or
# https://tuxmake.org/install-rpm/
# See https://docs.tuxmake.org/ for complete documentation.
# Original tuxmake command with fragments listed below.
# tuxmake --runtime podman --target-arch x86_64 --toolchain clang-nightly --kconfig defconfig --kconfig-add
https://raw.githubusercontent.com/Linaro/meta-lkft/kirkstone/meta/recipes-kernel/linux/files/systemd.config
--kconfig-add tools/testing/selftests/tc-testing/config LLVM=1 LLVM_IAS=1 debugkernel cpupower headers kernel kselftest
modules


tuxmake --runtime podman --target-arch x86_64 --toolchain clang-nightly --kconfig
https://storage.tuxsuite.com/public/tuxsuite/vishal/builds/2XQzJrgiVJMce65rDS5VZAFKw5w/config LLVM=1 LLVM_IAS=1
debugkernel cpupower headers kernel kselftest modules
```

Linaro

Local Test Reproducer

```
#!/bin/sh
# TuxRun is a command line tool and Python library that provides a way to
# boot and test linux kernels.
# TuxRun supports the concept of runtimes. For that to work it requires that
# you install podman or docker on your system.
# To install tuxrun to your home directory at ~/.local/bin:
# pip3 install -U --user tuxrun==0.49.2
# Or install a deb/rpm depending on the running distribution
# See https://tuxmake.org/install-deb/ or
# https://tuxmake.org/install-rpm/
# See https://tuxrun.org/ for complete documentation.
# Please follow the additional instructions if the tests are related to FVP:
# https://tuxrun.org/run-fvp/
#
tuxrun --runtime podman --device qemu-x86_64 --boot-args ''"'"'rw systemd.log_level=warning'"'"'' --kernel
https://storage.tuxsuite.com/public/tuxsuite/vishal/builds/2XQzJrgiVJMce65rDS5VZAFKw5w/bzImage --modules
https://storage.tuxsuite.com/public/tuxsuite/vishal/builds/2XQzJrgiVJMce65rDS5VZAFKw5w/modules.tar.xz --rootfs
https://storage.tuxboot.com/debian/bookworm/amd64/rootfs.ext4.xz --parameters
KSELFTEST=https://storage.tuxsuite.com/public/tuxsuite/vishal/builds/2XQzJrgiVJMce65rDS5VZAFKw5w/kselftest.tar.xz
--image docker.io/linaro/tuxrun-dispatcher:v0.49.2 --tests kselftest-tc-testing --timeouts boot=15
kselftest-tc-testing=20
```

TuxSuite Test patches

```
tuxsuite plan \
  https://gitlab.com/vishalbhoj/tc-testing/-/raw/main/tc-x86.yaml \
  --git-repo https://git.kernel.org/pub/scm/linux/kernel/git/netdev/net-next.git \
  --git-ref main \
  --patch-series https://lore.path-to.mbox

--patch-series file:///path/to/patches.tar.gz
--patch-series file:///path/to/a/directory/with/patches/
```

Linaro

For More Information

- tc-testing Demo https://gitlab.com/vishalbhoj/tc-testing
- For information on the TuxMake, TuxBake, and TuxRun open source projects see:
  - https://gitlab.com/Linaro/tuxmake
  - https://gitlab.com/Linaro/tuxbake
  - https://gitlab.com/Linaro/tuxrun

- Dashboard visualization:
  - https://qa-reports.linaro.org/~anders.roxell/tc-testing-demo/

- For information on the TuxSuite cloud service https://docs.tuxsuite.com/

- Signup for a trial at https://tuxsuite.com or email tuxsuite@linaro.org

- For more information, please contact us at tuxsuite@linaro.org

References

Plan URL:
https://tuxapi.tuxsuite.com/v1/groups/tuxsuite/projects/vishal/plans/2XQzJlLdfszm45lF6111htoaXcZ

Test Failure Reproducer:
https://tuxapi.tuxsuite.com/v1/groups/tuxsuite/projects/vishal/tests/2XQzJyq4aeMzrEuGbtNaU1ZKh51/tuxsuite_reproducer

Local Reproducer:
https://tuxapi.tuxsuite.com/v1/groups/tuxsuite/projects/vishal/tests/2XQzJyq4aeMzrEuGbtNaU1ZKh51/reproducer