TuxMake and TuxBuild

Dan Rue <dan.rue@linaro.org>
Antonio Terceiro <antonio.terceiro@linaro.org>

Hi, the build fails for me with the following error!

defconfig

gcc 9

mips.

I can’t reproduce. What kernel config did you use?

I can’t reproduce. What compiler?

I can’t reproduce. What architecture?

Linux Kernel Building Problem Statements

- Build environments are not reproducible
- Build environments are not portable
- Building the matrix of toolchains x architectures x configs x targets non-trivial
- Each toolchain, architecture, config, and target being built has user-space requirements
- Managing build artifacts can be messy
- Automating of the above is usually DIY and non-portable
- Making build artifacts available for testing is non-trivial
- Build coverage is generally constrained by hardware and integration effort

What if we could version and distribute build environments similarly to how we version Linux source code?

What if we weren’t constrained by build capacity?

Introducing TuxMake

- Open source, MIT licensed
- Thin wrapper around Linux kernel builds
- Python based command-line tool and library
- Provides portable and hermetic docker-based build environments
  - Pristine build containers contain kernel building requirements and nothing more; debian-based
- Provides common interface to building across architectures, toolchains, configs, and targets
- Collects artifacts in common location per-build; kernel source is read-only
- 100% unit and integration test coverage, requires only python core libraries
- Written and maintained by Antonio Terceiro <antonio.terceiro@linaro.org>

Opinionated Defaults - build everything; native arch

```
$ tuxmake
# command line: tuxmake
make --silent --keep-going --jobs=72 O=/home/dan.rue/.cache/tuxmake/builds/1/tmp defconfig
make --silent --keep-going --jobs=72 O=/home/dan.rue/.cache/tuxmake/builds/1/tmp
grep -q CONFIG_MODULES=y /home/dan.rue/.cache/tuxmake/builds/1/tmp/.config
make --silent --keep-going --jobs=72 O=/home/dan.rue/.cache/tuxmake/builds/1/tmp modules
make --silent --keep-going --jobs=72 O=/home/dan.rue/.cache/tuxmake/builds/1/tmp modules_i
tar caf /home/dan.rue/.cache/tuxmake/builds/1/tmp/modules.tar.gz -C /home/dan.rue/.cache/t
test -d arch/x86_64/boot/dts
# Skipping dtbs because precondition failed
I: config: PASS in 0:00:01.308105
I: kernel: PASS in 0:00:52.821837
I: modules: PASS in 0:00:02.641460
I: dtbs: SKIP in 0:00:00
I: build output in /home/dan.rue/.cache/tuxmake/builds/1
$
```

Build in Docker using ‘--runtime docker’

```
$ tuxmake --runtime docker
# command line: tuxmake --runtime docker
Using default tag: latest
latest: Pulling from tuxmake/x86_64_gcc
Digest: sha256:5a9fb76b7c5725725235e5d76b3e5cef0765a57a4db68fcdd1c2deb45638aa61
Status: Image is up to date for tuxmake/x86_64_gcc:latest
docker.io/tuxmake/x86_64_gcc:latest
make --silent --keep-going --jobs=72 O=/home/dan.rue/.cache/tuxmake/builds/2/tmp defconfig
make --silent --keep-going --jobs=72 O=/home/dan.rue/.cache/tuxmake/builds/2/tmp
```

- Requires docker to be available
- Each ‘make’ command happens transparently in an ephemeral docker container
- Source and output paths are mounted in; build runs as your uid
- x86_64 and arm64 native containers provided by default for each supported cross-build target architecture and toolchain

Build arm64 with clang-10 using Docker

```
$ tuxmake --runtime docker --toolchain clang-10 --target-arch arm64
# command line: tuxmake --runtime docker --toolchain clang-10 --target-arch arm64
Using default tag: latest
latest: Pulling from tuxmake/arm64_clang-10
Digest: sha256:e8128416588564c2a3294d8be1690799bd46b9349bbb7160e61a8a0dfc9d2023
Status: Image is up to date for tuxmake/arm64_clang-10:latest
docker.io/tuxmake/arm64_clang-10:latest
make --silent --keep-going --jobs=72 O=/home/dan.rue/.cache/tuxmake/builds/3/tmp ARCH=arm64 CROSS_CO
MPILE=aarch64-linux-gnu- HOSTCC=clang-10 CC=clang-10 defconfig
make --silent --keep-going --jobs=72 O=/home/dan.rue/.cache/tuxmake/builds/3/tmp ARCH=arm64 CROSS_CO
MPILE=aarch64-linux-gnu- HOSTCC=clang-10 CC=clang-10
```

Build artifacts are saved in ~/.cache/tuxmake/

```
$ ls /home/dan.rue/.cache/tuxmake/builds/1
build.log  bzImage  config  metadata.json  modules.tar.gz
$ head -15 metadata.json
{
    "build": {
        "targets": [
            "config",
            "kernel",
            "modules",
            "dtbs"
        ],
        "target_arch": "x86_64",
        "toolchain": "gcc",
        "wrapper": "none",
        "environment": {},
        "kconfig": "defconfig",
        "kconfig_add": [],
        "jobs": 72,
$
```

Supported architecture/toolchain combinations

```
$ tuxmake --print-support-matrix
      arc    arm    arm64  i386   mips   riscv  x86_64
clang yes    yes    yes    yes    yes    yes    yes
gcc   yes    yes    yes    yes    yes    yes    yes
$ tuxmake --print-support-matrix --runtime docker
         arc    arm    arm64  i386   mips   riscv  x86_64
clang    no     yes    yes    yes    yes    yes    yes
clang-10 no     yes    yes    yes    no     yes    yes
clang-8  no     yes    yes    yes    no     yes    yes
clang-9  no     yes    yes    yes    no     yes    yes
gcc      yes    yes    yes    yes    yes    yes    yes
gcc-10   no     yes    yes    yes    no     yes    yes
gcc-8    no     yes    yes    yes    yes    yes    yes
gcc-9    yes    yes    yes    yes    no     no     yes
$
```

Hi, the following build fails for me.

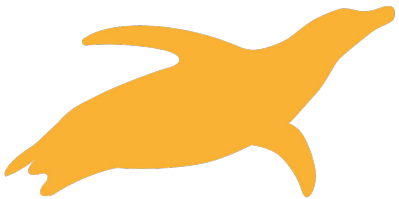
tuxmake -r docker -a arm64 -t clang-10 -k defconfig

Getting Started

- Current status: Alpha
- ‘pip3 install tuxmake’
- Debian packages coming soon
- Project location: https://gitlab.com/Linaro/tuxmake
- Feature requests: https://gitlab.com/Linaro/tuxmake/-/issues

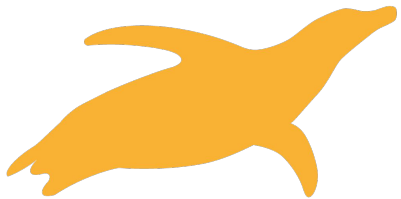
Introducing TuxBuild

- Fully managed service by Linaro
- Open source command-line client; closed-source back end
- Solves the build capacity problem
- Provides TuxMake “as-a-service”
- Cloud-native highly concurrent Linux kernel building service
- Easy to integrate with CI or developer workflows
- Used in production by LKFT since January 2020
- Supports individual builds as well as build sets
- Interface and features similar to TuxMake’s
  - gcc 8/9/10, clang 8/9/10
  - Arm64, arm, x86_64, i386, mips, arc, riscv

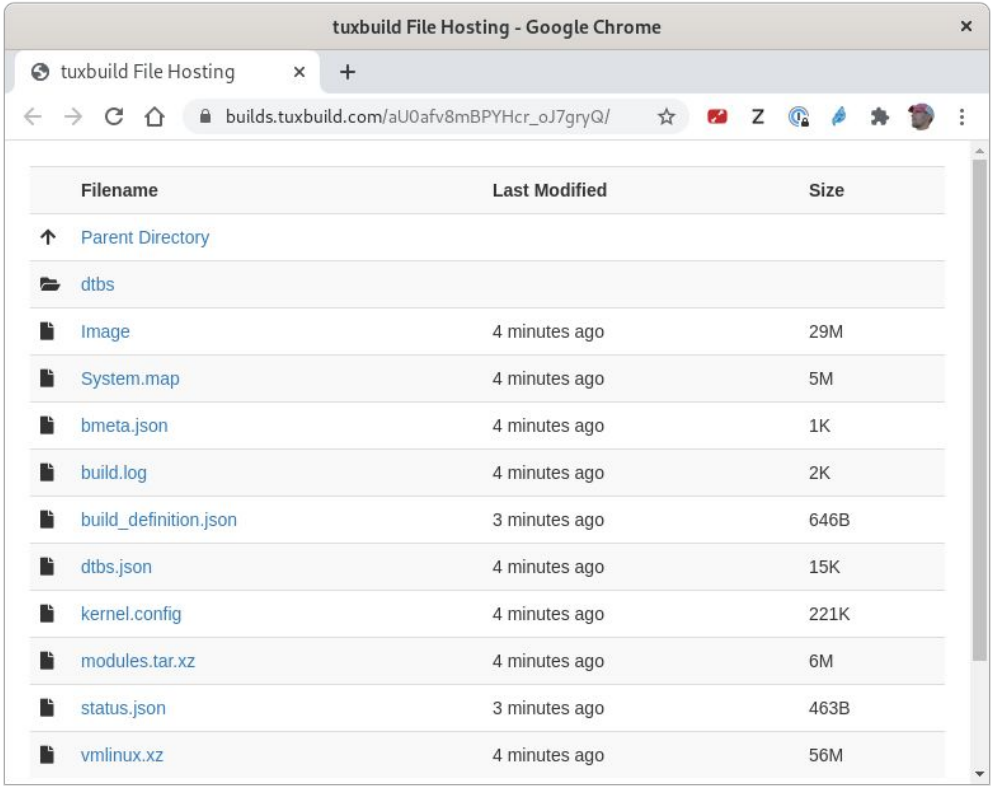


TuxBuild Example

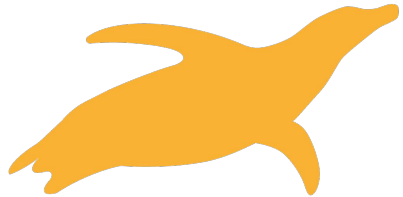
```
$ tuxbuild build --git-repo 'https://git.kernel.org/pub/scm/linux/kernel/git/torvalds/linux.git' --git-ref master --target-arch arm64 --kconfig defconfig --toolchain clang-10
Building Linux Kernel https://git.kernel.org/pub/scm/linux/kernel/git/torvalds/linux.git at master
⏳ Queued:  arm64 (defconfig) with clang-10 @ https://builds.tuxbuild.com/aU0afv8mBPYHcr_oJ7gryQ/
⚗️ Building: 7eac66d0456f ("Merge tag 'vfio-v5.9-rc2' of git://github.com/awilliam/linux-vfio") arm64 (defconfig) with clang-10 @ https://builds.tuxbuild.com/aU0afv8mBPYHcr_oJ7gryQ/
👾 Pass (8 warnings): 7eac66d0456f ("Merge tag 'vfio-v5.9-rc2' of git://github.com/awilliam/linux-vfio") arm64 (defconfig) with clang-10 @ https://builds.tuxbuild.com/aU0afv8mBPYHcr_oJ7gryQ/
```



TuxBuild Artifact Hosting

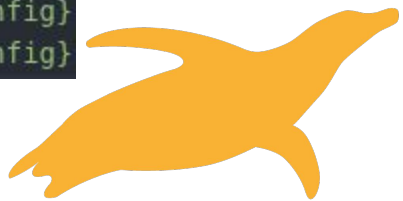


Filename	Last Modified	Size
Parent Directory		
dtbs		
Image	4 minutes ago	29M
System.map	4 minutes ago	5M
bmeta.json	4 minutes ago	1K
build.log	4 minutes ago	2K
build_definition.json	3 minutes ago	646B
dtbs.json	4 minutes ago	15K
kernel.config	4 minutes ago	221K
modules.tar.xz	4 minutes ago	6M
status.json	3 minutes ago	463B
vmlinux.xz	4 minutes ago	56M



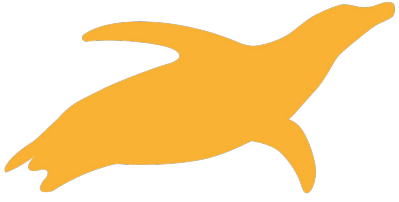
Build Set Definition - can be local or a URL

```yaml
sets:
  - name: basic
    builds:
      - {target_arch: arm,    toolchain: gcc-9,    kconfig: tinyconfig}
      - {target_arch: arm,    toolchain: gcc-10,   kconfig: tinyconfig}
      - {target_arch: arm,    toolchain: clang-10, kconfig: tinyconfig}
      - {target_arch: arm64,  toolchain: gcc-9,    kconfig: tinyconfig}
      - {target_arch: arm64,  toolchain: gcc-10,   kconfig: tinyconfig}
      - {target_arch: arm64,  toolchain: clang-10, kconfig: tinyconfig}
      - {target_arch: i386,   toolchain: gcc-9,    kconfig: tinyconfig}
      - {target_arch: i386,   toolchain: gcc-10,   kconfig: tinyconfig}
      - {target_arch: x86_64, toolchain: gcc-9,    kconfig: tinyconfig}
      - {target_arch: x86_64, toolchain: gcc-10,   kconfig: tinyconfig}
      - {target_arch: x86_64, toolchain: clang-10, kconfig: tinyconfig}
      - {target_arch: arc,    toolchain: gcc-10,   kconfig: tinyconfig}
      - {target_arch: riscv,  toolchain: gcc-10,   kconfig: tinyconfig}
      - {target_arch: mips,   toolchain: gcc-10,   kconfig: tinyconfig}
```

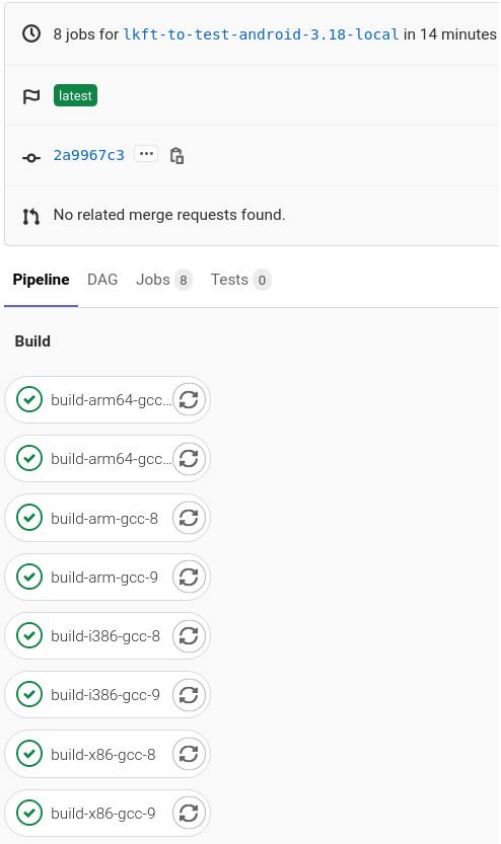


Build Set Definition - can be local or a URL

```
$ tuxbuild build-set --tux-config basic.yaml --set-name basic --json-out buildset.json --git-repo 'https://
Building Linux Kernel build set basic
⏳ Queued:  arm (tinyconfig) with gcc-9 @ https://builds.tuxbuild.com/Weyb7S-yyVigitcx9WslnQ/
⏳ Queued:  arm (tinyconfig) with gcc-10 @ https://builds.tuxbuild.com/DyNVPt2NBkv-NlyS8wMZCw/
⏳ Queued:  arm (tinyconfig) with clang-10 @ https://builds.tuxbuild.com/pzzQIxE3YRlfnoaonx1suA/
⏳ Queued:  arm64 (tinyconfig) with gcc-9 @ https://builds.tuxbuild.com/-s5TrcJeaFKS1hjoWqfNUA/
⏳ Queued:  arm64 (tinyconfig) with gcc-10 @ https://builds.tuxbuild.com/cjXcbZxtLXYVfNREugSWkA/
⏳ Queued:  arm64 (tinyconfig) with clang-10 @ https://builds.tuxbuild.com/Pca0QyMvNuk_sxKxjzq_JQ/
⏳ Queued:  i386 (tinyconfig) with gcc-9 @ https://builds.tuxbuild.com/LN62wDFOeDvPAPSIgpj9Lw/
⏳ Queued:  i386 (tinyconfig) with gcc-10 @ https://builds.tuxbuild.com/OHPhqbH0kEmEEx1v26Xflw/
⏳ Queued:  x86_64 (tinyconfig) with gcc-9 @ https://builds.tuxbuild.com/Q_0nauNS9zowrtxME7gs6Q/
⏳ Queued:  x86_64 (tinyconfig) with gcc-10 @ https://builds.tuxbuild.com/YrMJ6j2Gl6Xk6jv5RpO9zw/
⏳ Queued:  x86_64 (tinyconfig) with clang-10 @ https://builds.tuxbuild.com/sB7_f8kKpo2Pn3ydS2CxCQ/
⏳ Queued:  arc (tinyconfig) with gcc-10 @ https://builds.tuxbuild.com/Bd7r4KjVv4s-iOpU0ueEBQ/
⏳ Queued:  riscv (tinyconfig) with gcc-10 @ https://builds.tuxbuild.com/cIx8tKOG111Hg8C2mk7D2g/
⏳ Queued:  mips (tinyconfig) with gcc-10 @ https://builds.tuxbuild.com/RU7lvZwLt85MYvqpois6oA/
🚧 Building: 7eac66d0456f ("Merge tag 'vfio-v5.9-rc2' of git://github.com/awilliam/linux-vfio") mips (tinyc
🚧 Building: 7eac66d0456f ("Merge tag 'vfio-v5.9-rc2' of git://github.com/awilliam/linux-vfio") x86_64 (tin
🚧 Building: 7eac66d0456f ("Merge tag 'vfio-v5.9-rc2' of git://github.com/awilliam/linux-vfio") x86_64 (tin
🚧 Building: 7eac66d0456f ("Merge tag 'vfio-v5.9-rc2' of git://github.com/awilliam/linux-vfio") arm (tinyco
🚧 Building: 7eac66d0456f ("Merge tag 'vfio-v5.9-rc2' of git://github.com/awilliam/linux-vfio") arm (tinyco
🚧 Building: 7eac66d0456f ("Merge tag 'vfio-v5.9-rc2' of git://github.com/awilliam/linux-vfio") x86_64 (tin
🚧 Building: 7eac66d0456f ("Merge tag 'vfio-v5.9-rc2' of git://github.com/awilliam/linux-vfio") arc (tinyco
🚧 Building: 7eac66d0456f ("Merge tag 'vfio-v5.9-rc2' of git://github.com/awilliam/linux-vfio") arm64 (tiny
🚧 Building: 7eac66d0456f ("Merge tag 'vfio-v5.9-rc2' of git://github.com/awilliam/linux-vfio") i386 (tinyc
🚧 Building: 7eac66d0456f ("Merge tag 'vfio-v5.9-rc2' of git://github.com/awilliam/linux-vfio") arm64 (tiny
🚧 Building: 7eac66d0456f ("Merge tag 'vfio-v5.9-rc2' of git://github.com/awilliam/linux-vfio") i386 (tinyc
🚧 Building: 7eac66d0456f ("Merge tag 'vfio-v5.9-rc2' of git://github.com/awilliam/linux-vfio") arm (tinyco
🚧 Building: 7eac66d0456f ("Merge tag 'vfio-v5.9-rc2' of git://github.com/awilliam/linux-vfio") arm64 (tiny
🚧 Building: 7eac66d0456f ("Merge tag 'vfio-v5.9-rc2' of git://github.com/awilliam/linux-vfio") riscv (tiny
🎉 Pass (1 warning): 7eac66d0456f ("Merge tag 'vfio-v5.9-rc2' of git://github.com/awilliam/linux-vfio") x86
🎉 Pass: 7eac66d0456f ("Merge tag 'vfio-v5.9-rc2' of git://github.com/awilliam/linux-vfio") i386 (tinyconfi
🎉 Pass: 7eac66d0456f ("Merge tag 'vfio-v5.9-rc2' of git://github.com/awilliam/linux-vfio") arm64 (tinycon
🎉 Pass: 7eac66d0456f ("Merge tag 'vfio-v5.9-rc2' of git://github.com/awilliam/linux-vfio") arm64 (tinycon
🎉 Pass: 7eac66d0456f ("Merge tag 'vfio-v5.9-rc2' of git://github.com/awilliam/linux-vfio") arc (tinyconfig
🎉 Pass: 7eac66d0456f ("Merge tag 'vfio-v5.9-rc2' of git://github.com/awilliam/linux-vfio") i386 (tinyconfi
🎉 Pass: 7eac66d0456f ("Merge tag 'vfio-v5.9-rc2' of git://github.com/awilliam/linux-vfio") arm (tinyconfig
🎉 Pass: 7eac66d0456f ("Merge tag 'vfio-v5.9-rc2' of git://github.com/awilliam/linux-vfio") riscv (tinycon
🎉 Pass: 7eac66d0456f ("Merge tag 'vfio-v5.9-rc2' of git://github.com/awilliam/linux-vfio") arm (tinyconfig
🎉 Pass: 7eac66d0456f ("Merge tag 'vfio-v5.9-rc2' of git://github.com/awilliam/linux-vfio") mips (tinyconfi
🎉 Pass: 7eac66d0456f ("Merge tag 'vfio-v5.9-rc2' of git://github.com/awilliam/linux-vfio") arm64 (tinycon
🎉 Pass (1 warning): 7eac66d0456f ("Merge tag 'vfio-v5.9-rc2' of git://github.com/awilliam/linux-vfio") x86
🎉 Pass: 7eac66d0456f ("Merge tag 'vfio-v5.9-rc2' of git://github.com/awilliam/linux-vfio") arm (tinyconfig
🎉 Pass (1 warning): 7eac66d0456f ("Merge tag 'vfio-v5.9-rc2' of git://github.com/awilliam/linux-vfio") x86
$
```



Kernel Maintainer Build Pipeline in GitLab



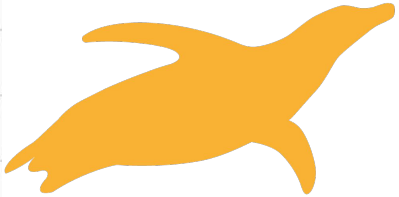
- Automatically builds after a git push
- 268 builds, including allmodconfig, allnoconfig for 4 architectures
- Completes in 14-16 minutes

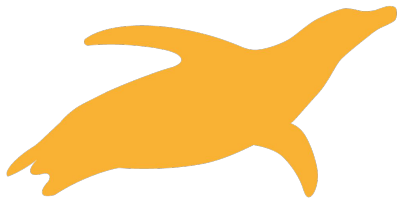
https://gitlab.com/Linaro/lkft/users/lee.jones/lag-linaro-linux/-/pipelines

LKFT Pipeline

- LKFT builds kernels with TuxBuild and then tests them on hardware
- Performs ~70 builds in ~15 minutes

https://lkft.linaro.org/



Getting Started

- TuxBuild is a fully managed service by Linaro
- TuxBuild Status: Early Access
  - Reach out to us at tuxbuild@linaro.org to be added to our waiting list.
- More information and TuxBuild CLI can be found at https://gitlab.com/Linaro/tuxbuild