



Building Yocto Project at scale with TuxSuite

Vishal Bhoj, Linaro

Yocto Project Summit, 2022.05

Agenda

- About Linaro and me
- What is TuxSuite ?
- TuxSuite Usecases last year
- TuxSuite YP/OE support
- Build definition Template
- TuxBake
- Building a single YP/OE distro image with TuxSuite
- Building multiple YP/OE distro images with TuxSuite Plans

Linaro

- Driving open source software development on Arm since 2010, providing the tools, Linux kernel quality and security needed for a solid foundation to innovate on.
- Working with member companies and the open source community to maintain the Arm software ecosystem and enabling new markets on Arm architecture.
- Linaro Developer Services: Helping build, deploy and maintain your products on Arm
- Software Services that accelerate development and testing of open source Operating System software.
- Linaro projects:
 - Linux kernel
 - toolchains
 - security
 - continuous integration testing
 - Automotive
 - IoT & Edge Devices, Client Devices
 - Cloud Computing & Servers.

About Me

- Senior Engineer in Linaro as part of Tuxsuite Team, developing cloud solutions to build various open source software products.
- 14 years of experience designing and building CI/CD solutions for a variety of embedded software stacks.
- Have worked on various components of the stack
 - u-boot
 - edk2
 - AOSP
 - Yocto

What is TuxSuite?

- On-demand APIs and tools executing in parallel at scale in the cloud.
 - Building Linux kernels
 - Testing Linux kernels on Qemu
 - Building Yocto Project / OpenEmbedded and derivatives
- TuxSuite can scale to more than 1000 builds or tests in parallel within minutes
- [tuxsuite cli](#) is a python cli application to talk to the API endpoint
- Easily Reproducible in the cloud and locally thanks to OSS tools:
 - Linux Kernel - [TuxMake](#)
 - Boot Testing with Qemu - [TuxRun](#)
 - Build YP/OE - [TuxBake](#)

TuxSuite Usecases Last Year

- LKFT
 - Linux Kernel build, boot and LTP tests
 - OE builds for building userspace for kernel testing
 - Peak usage in a day
 - 1 2000+ hrs \approx 40+ 8 core machines doing the builds for 24 hrs.
 - 2 9500+ builds
 - 136,995 builds in April 2022
 - 1,203,113 kernel built in 2021
- Individual Linux Kernel maintainers use it to build test
- ClangbuiltLinux - Linux kernel build tests with different Clang versions
 - Peak Usage in a day: 1000+ hrs for 2000+ builds
 - 54,043 builds in the month of April 2022
 - More details are available here: [Deepdive Into ClangBuiltLinux](#)
- Morello - Running Linux and Android tests on FVP in the cloud

TuxSuite Applications

- Easily integrated with any modern CI systems and Code Review tools:
 - Gerrit
 - Gitlab CI
 - Github Actions
 - Jenkins
 - Travis CI
- Developers can use it for wider build/boot test locally

TuxSuite YP/OE Support

- TuxSuite supports building YP/OE
 - Supported backends:
 - [TuxBake](#)
 - [kas](#)
- Per project caching:
 - shared SSTATE Mirror.
 - shared DL_DIR.

TuxSuite YP/OE Support

- Publish build artifacts at a non-guessable URL

Filename	Last Modified	Size
↑ Parent Directory		
📁 images/		
📁 licenses/		
📁 rpm/		
📄 bitbake-environment	2 hours ago	1M
📄 build-debug.log	2 hours ago	498K
📄 build-definition.json	2 hours ago	806B
📄 build.log	2 hours ago	454K

[tuxpub 0.1.10](#), by [Linaro](#).

Tip: If `?export=json` is appended to the URL, a json representation of the page will be provided.

TuxSuite + TuxBake

- TuxBake is an Open Source Python3 Application to build YP/OE.
- Containerized build environment.
 - Ubuntu-<16.04/18.04/20.04/22.04>
 - Debian-<buster/bullseye/stretch>
 - Centos-8
 - Fedora-<33/34>
 - opensuse-leap-<15.1/15.2>
- Uses Docker/Podman for providing the containerized environment.
- Uses build-definition Template.
- Git / Android Repo tool to manage sources.
- Allows users to define local.conf, bblayers.conf, environment variables, sstate_mirror, dl_dir, distro, machine, container.

Build definition fields

- **sources** (dictionary with a single item): could be one `git_trees`, `repo` or `kas`
 - a. **git_trees** (list of dictionaries): url, branch and optionally commit ids/refs
 - b. **repo** (a dictionary): branch, manifest file name and manifest repo url
 - c. **kas** (a dictionary): url, branch, kas yaml and optionally ref.
- **envsetup**: path to the script, relative to the source directory, sourced to setup bitbake environment
- **distro, machine, target**: variables passed to the bitbake command
- **container**: the container used by Docker to do the build.
- **local_conf, bblayers_conf** (list of strings): each entry corresponds to a line in local.conf file.
- **environment** (dictionary of environment variables)

build-definition examples - Yocto Project / Poky

```
$ # Install tuxsuite
$ pip3 install tuxsuite
$ cat yocto.json
{
  "container": "ubuntu-20.04",
  "envsetup": "poky/oe-init-build-env",
  "distro": "poky",
  "machine": "qemux86-64",
  "target": "core-image-minimal",
  "sources": {
    "git_trees": [
      {
        "url": "git://git.yoctoproject.org/poky",
        "branch": "kirkstone",
        "sha": "780eeec8851950ee6ac07a2a398ba937206bd2e4"
      }
    ]
  }
}
```

build-definition examples - Yocto Project / Poky

```
$ tuxsuite bake submit yocto.json
```

```
*** WARNING: BITBAKE SUPPORT IS EXPERIMENTAL ***
```

```
Building target: core-image-minimal with bitbake from {'git_trees': [{'url':  
'git://git.yoctoproject.org/poky', 'branch': 'honister'}]} source with distro:  
poky machine: qemux86-64 arguments  
uid: 291KafvkrqBvDt8iF03nf2jQWjS
```

```
⚙️ Provisioning: {'git_trees': [{'url': 'git://git.yoctoproject.org/poky',  
'branch': 'honister'}]} with poky qemux86-64 for core-image-minimal:  
https://oebuilds.tuxbuild.com/291KafvkrqBvDt8iF03nf2jQWjS/
```

```
🚀 Running: {'git_trees': [{'url': 'git://git.yoctoproject.org/poky', 'branch':  
'honister'}]} with poky qemux86-64 for core-image-minimal:  
https://oebuilds.tuxbuild.com/291KafvkrqBvDt8iF03nf2jQWjS/
```

```
🎉 Pass: {'git_trees': [{'url': 'git://git.yoctoproject.org/poky', 'branch':  
'honister'}]} with poky qemux86-64 for core-image-minimal:  
https://oebuilds.tuxbuild.com/291KafvkrqBvDt8iF03nf2jQWjS/
```

build-definition examples - Oniro

```
$ cat oniro.json
{
  "container": "ubuntu-20.04",
  "distro": "oniro-linux",
  "environment": {
    "TEMPLATECONF": "../oniro/flavours/linux"
  },
  "envsetup": "oe-core/oe-init-build-env",
  "machine": "qemux86-64",
  "sources": {
    "repo": {
      "branch": "kirkstone",
      "manifest": "default.xml",
      "url": "https://gitlab.eclipse.org/eclipse/oniro-core/oniro"
    }
  },
  "target": "oniro-image-base-tests"
}
```

build-definition examples - Oniro

```
$ tuxsuite bake submit oniro.json
```

```
*** WARNING: BITBAKE SUPPORT IS EXPERIMENTAL ***
```

```
Building target: oniro-image-base with bitbake from {'repo': {'branch': 'kirkstone',  
'manifest': 'default.xml', 'url': 'https://booting.oniroproject.org/distro/oniro'}}  
source with distro: oniro-linux machine: qemux86-64 arguments  
uid: 291SyRebaz5UNPVoaWWDbwWtecI
```

```
⚙ Provisioning: {'repo': {'branch': 'kirkstone', 'manifest': 'default.xml', 'url':  
'https://booting.oniroproject.org/distro/oniro'}} with oniro-linux qemux86-64 for  
oniro-image-base: https://oebuilds.tuxbuild.com/291SyRebaz5UNPVoaWWDbwWtecI/
```

```
🚀 Running: {'repo': {'branch': 'kirkstone', 'manifest': 'default.xml', 'url':  
'https://booting.oniroproject.org/distro/oniro'}} with oniro-linux qemux86-64 for  
oniro-image-base: https://oebuilds.tuxbuild.com/291SyRebaz5UNPVoaWWDbwWtecI/
```

```
🎉 Pass: {'repo': {'branch': 'kirkstone', 'manifest': 'default.xml', 'url':  
'https://booting.oniroproject.org/distro/oniro'}} with oniro-linux qemux86-64 for  
oniro-image-base: https://oebuilds.tuxbuild.com/291SyRebaz5UNPVoaWWDbwWtecI/
```

build-definition examples - Kas

```
$ cat kas.json
{
  "sources": {
    "kas": {
      "url": "https://git.codelinaro.org/linaro/dependable-boot/meta-ts",
      "yaml": "ci/zynqmp-starter.yml"
    }
  }
}
```


build-definition examples - Kas

```
$ tuxsuite bake submit kas.json
```

```
*** WARNING: BITBAKE SUPPORT IS EXPERIMENTAL ***
```

```
Building target: None with bitbake from {'kas': {'url':  
'https://git.codelinaro.org/linaro/dependable-boot/meta-ts', 'yaml':  
'ci/zynqmp-starter.yml'}} source with distro: None machine: None arguments  
uid: 291T4fWMaux9zNIZdAVQAFwlm8s
```

```
⚙ Provisioning: {'kas': {'url':  
'https://git.codelinaro.org/linaro/dependable-boot/meta-ts', 'yaml':  
'ci/zynqmp-starter.yml'}} with None None for None:  
https://oebuilds.tuxbuild.com/291T4fWMaux9zNIZdAVQAFwlm8s/
```

```
🚀 Running: {'kas': {'url': 'https://git.codelinaro.org/linaro/dependable-boot/meta-ts',  
'yaml': 'ci/zynqmp-starter.yml'}} with None None for None:  
https://oebuilds.tuxbuild.com/291T4fWMaux9zNIZdAVQAFwlm8s/
```

```
🎉 Pass: {'kas': {'url': 'https://git.codelinaro.org/linaro/dependable-boot/meta-ts',  
'yaml': 'ci/zynqmp-starter.yml'}} with None None for None:  
https://oebuilds.tuxbuild.com/291T4fWMaux9zNIZdAVQAFwlm8s/
```

TuxSuite Plans for Yocto - Poky master-next

```
$ cat poky-next.yaml
common: &commondata
  "container": "ubuntu-20.04"
  "distro": "rpb"
  "envsetup": "setup-environment"
  "sources": {
    "git_trees": [
      {
        "url":
"git://git.yoctoproject.org/poky",
        "branch": "master-next"
      }
    ]
  }
  "target": "core-image-minimal"
version: 1
name: Poky next validation
```

```
description: Build Poky for various qemu
jobs:
- name: next-tracking
  bakes:
  - { <<: *commondata, "machine": "qemuarm64" }
  - { <<: *commondata, "machine": "qemuarm" }
  - { <<: *commondata, "machine": "qemuarmv5" }
  - { <<: *commondata, "machine": "qemumips64" }
  - { <<: *commondata, "machine": "qemumips" }
  - { <<: *commondata, "machine": "qemuppc64" }
  - { <<: *commondata, "machine": "qemuppc" }
  - { <<: *commondata, "machine": "qemuriscv32" }
  - { <<: *commondata, "machine": "qemuriscv64" }
  - { <<: *commondata, "machine": "qemux86-64" }
  - { <<: *commondata, "machine": "qemux86" }
```

TuxSuite Plans for Yocto

```
$ tuxsuite plan poky-next.yaml
*** WARNING: BITBAKE SUPPORT IS EXPERIMENTAL ***
Running Bake plan 'Poky next validation': 'Build Poky for various qemu'
Plan https://tuxapi.tuxsuite.com/v1/groups/tuxsuite/projects/vishal/plans/2974WYyncHQsrfgwhdDvIElLOut

uid: 2974WYyncHQsrfgwhdDvIElLOut
⚙️ Provisioning: {'git_trees': [{'url': 'git://git.yoctoproject.org/poky', 'branch': 'master-next'}]} with
poky qemuarm64 for core-image-minimal: https://oebuilds.tuxbuild.com/2974WfMW5ffFuWdMvPepMHdr0BQ/
⚙️ Provisioning: {'git_trees': [{'url': 'git://git.yoctoproject.org/poky', 'branch': 'master-next'}]} with
poky qemuarm for core-image-minimal: https://oebuilds.tuxbuild.com/2974WhiciLNQzqRy355y0Foeey/
.
.
🚀 Running: {'git_trees': [{'url': 'git://git.yoctoproject.org/poky', 'branch': 'master-next'}]} with poky
qemuarm64 for core-image-minimal: https://oebuilds.tuxbuild.com/2974WfMW5ffFuWdMvPepMHdr0BQ/
🚀 Running: {'git_trees': [{'url': 'git://git.yoctoproject.org/poky', 'branch': 'master-next'}]} with poky
qemuarm for core-image-minimal: https://oebuilds.tuxbuild.com/2974WhiciLNQzqRy355y0Foeey/
.
.
```

TuxSuite Plans for Yocto

```
🚀 Running: {'git_trees': [{'url': 'git://git.yoctoproject.org/poky', 'branch': 'master-next'}]} with poky qemuarm for
core-image-minimal: https://oebuilds.tuxbuild.com/2974WhiciLNQzggqRy355y0FoeEY/
.
.
🎉 Pass: {'git_trees': [{'url': 'git://git.yoctoproject.org/poky', 'branch': 'master-next'}]} with poky qemuarmv5 for
core-image-minimal: https://oebuilds.tuxbuild.com/2974Wbe9Gen3HeXg4Px4Br7mHyy/
🎉 Pass: {'git_trees': [{'url': 'git://git.yoctoproject.org/poky', 'branch': 'master-next'}]} with poky qemumips for
core-image-minimal: https://oebuilds.tuxbuild.com/2974WchEEacRybPXsXAju9D9Z4W/
.
.
Summary: https://tuxapi.tuxsuite.com/v1/groups/tuxsuite/projects/vishal/plans/2974WYynchQsrfgwhdDvIElLOut
2974WfMW5ffFuWdMvPepMHdr0BQ 🎉 Pass
2974WhiciLNQzggqRy355y0FoeEY 🎉 Pass
2974Wbe9Gen3HeXg4Px4Br7mHyy 🎉 Pass
2974WbxRWEWpDQ2ZVoKAhGG45xR 🎉 Pass
2974WchEEacRybPXsXAju9D9Z4W 🎉 Pass
2974WhDS5h1ao2m1FbRHERVEcIC 🎉 Pass
2974WeGXcIU2FkXdkM0CHDZ0Ri7 🎉 Pass
2974Wekt2wuoUB4iakkdJOQfHhd 🎉 Pass
2974Wb8xdKq25w5kh4eehwJiGWM 🎉 Pass
2974WhhEG72AT3msg9tzJsDWLIG 🎉 Pass
2974WgTngYzmHY2B1ap7hffnW57 🎉 Pass
```

TuxSuite Applications - Yocto

- Layers premerge testing and tracking changes with Yocto upstream.
 - meta-rpb premerge testing: <https://github.com/96boards/oe-rpb-manifest/blob/qcom/master/github/workflows/premerge-template.yml>
 - upstream Poky master-next tracking: <https://ci.linaro.org/job/lt-qcom-poky-master/>
 - meta-qcom layer CI: <https://github.com/ndechesne/meta-qcom/tree/master/github/workflows>

Reproducing the build with TuxBake locally

```
$ tuxbake --build-definition yocto.json --src-dir $PWD/yocto
Cloning into 'poky'...
remote: Enumerating objects: 561369, done.
remote: Counting objects: 100% (1355/1355), done.
remote: Compressing objects: 100% (911/911), done.
remote: Total 561369 (delta 742), reused 685 (delta 425), pack-reused 560014
Receiving objects: 100% (561369/561369), 182.99 MiB | 524.00 KiB/s, done.
Resolving deltas: 100% (407297/407297), done.
Using default tag: latest
latest: Pulling from tuxbake/ubuntu-20.04
Digest: sha256:a4e2a7f852b53c5ba741ef0e4d459913df9a2135d0444d6ec8e211792c2ede00
Status: Image is up to date for tuxbake/ubuntu-20.04:latest
docker.io/tuxbake/ubuntu-20.04:latest
usermod: no changes
bash -c 'rm build/conf/bblayers.conf; rm build/conf/local.conf; source poky/oe-init-build-env build; cat
../extra_local.conf >> conf/local.conf; cat ../bblayers.conf >> conf/bblayers.conf || true; echo
''''Dumping local.conf.'''''; cat conf/local.conf; bitbake -e > bitbake-environment; bitbake
core-image-minimal'
rm: cannot remove 'build/conf/bblayers.conf': No such file or directory
rm: cannot remove 'build/conf/local.conf': No such file or directory
You had no conf/local.conf file. This configuration file has therefore been
created for you with some default values. You may wish to edit it to, for
example, select a different MACHINE (target hardware). See conf/local.conf
for more information as common configuration options are commented.
.
.
```

TuxSuite Yocto Roadmap

- Boot Test Yocto images on Qemu with TuxTest
- no-cache option to skip sstate_cache usage
- Support to download sstate_cache objects
- Serve sstate_cache and dl_dir from TuxSuite Cloud to tuxbake
- artifacts download by CLI
- Publish Statistics for the build

TuxSuite Token

We have trial vouchers available for those interested in experimentation with the build service.

Please request a tuxsuite token via the: [TuxSuite Token Request](#)
Getting Started with TuxSuite: <https://learn.tuxsuite.com>

TuxSuite References

- <https://tuxsuite.com>
- <https://tuxrun.org>
- <https://tuxmake.org>
- <https://learn.tuxsuite.com>
- <https://gitlab.com/Linaro/tuxsuite>
- <https://gitlab.com/Linaro/tuxbake>
- <https://gitlab.com/Linaro/tuxmake>
- <https://gitlab.com/Linaro/tuxrun>
- <https://gitlab.com/Linaro/tuxsuite/-/tree/master/examples/bitbake>



yocto
PROJECT

THE
LINUX
FOUNDATION