Living on Master:

Using Yocto Project, Jenkins and LAVA for a rolling release

Tim Orling
Intel Open Source Technology Center
You’re doing it all wrong.

Here's why.
Traditional Approach: “Fork It and Forget It”

- Pin upstream at a fixed version
- Make changes in tree
- No patches sent upstream
Traditional Approach: “Fork It and Forget It”

- Obsolete from day one
- Upstream never stops
- Huge technical debt
Follow the Linux* kernel release model

- http://www.kroah.com/log/blog/2018/02/05/linux-kernel-release-model/
- Upstream First
- Backport to stable branches
- Constantly moving code-base
Everyone has their own solution

Time to share.
Motivations

Open Source

Enterprise Class

Off the Shelf

Distributed
Inspirations

- Ostro
- IoT Refkit
- kernelci.org
- ELC(E) Hallway Track
System overview
Repository layout
Use automation to stay up to date

- Continuous Integration job runs daily

1. autosync git submodules
2. git review [pull request]
3. trigger build
Testing production images

- Tests run on every build

pytest is an Open Source testing framework
DUT = Device Under Test
LAN = Local Area Network
PDU = Power Distribution Unit
Testing System

DUT = Device Under Test
LAN = Local Area Network
PDU = Power Distribution Unit
Next Steps

- Test Case events working well
- Measurements and Units under development
- Need to complete internal Open Sourcing process
Next Steps

- Industry Standard Result Formats
  - Test Anything Protocol
  - JUnit.xml

LAVA dashboard
Next Steps

Data Analysis and Visualization

Elastic Stack

Elasticsearch  Logstash  Kibana
Let’s do this.

Moving forward.
Reference Enterprise Class System

End-to-End Complete
- Orchestrated containers?
  - “Easy” to deploy
  - “Effortless” to replicate
- Internal Layer Index
- Jenkins Master
- Artifact server
- Data Visualization

Public LAVA server
- Share and Re-use test definitions
- Distributed LAVA dispatchers
  - Containerized
  - Better HW coverage
    - Each site focuses on platforms they care about
Questions?
Reference

- https://validation.linaro.org
- https://github.com/Linaro/lava-server
- https://github.com/Linaro/lava-dispatcher
- https://pytest.org
- https://github.com/pytest-dev
- https://git.linaro.org/lava-team/refactoring.git/
- https://kernelci.org
- https://jenkins.io
- https://www.elastic.co/
- https://lists.yoctoproject.org/listinfo/automated-testing
Thank you.

Special Thanks to:
- John Akre
- Kevron Rees
- Randy Witt
- Brian Avery
- Stephano Cetola