Yocto Project® CI/CD Developer Workflow

Konrad Scherer, Wind River

Yocto Project Dev Day Virtual, North America, 2020
Introduction

- Linux Infrastructure Lead at Wind River
- Maintain 100+ servers across 3 DC
- Current Challenge is improving Developer Productivity
- Konrad.Scherer@windriver.com
Agenda

- Wrigel Build System
- Yocto Developer Workflow Challenges
- DevBuilds and Build Failure Login
- Future Work
Wrigel the Third

- Docker, Docker Swarm and Jenkins
- Designed for easy setup of small cluster
- Independent of Wind River network
- Internal builds and in AWS

https://github.com/WindRiver-OpenSourceLabs/ci-scripts
Developer Workflow Challenges

- "GitOps" tools assume a single Git repository
- How to manage collections of 40+ layers?
- How to support changeset across multiple repos?
- How to manage write access to git repositories?
- What configurations to build and test for a changeset?
- Shorten feedback loops and minimize co-ordination
WR setup tool and git-repo

- Git-repo manages large tree of repos
- Manifest generation by setup tool
- Pro: Git-repo detects changes across entire tree
- Pro: Automatic rebase of local commits when syncing
- Con: Fork with support for bare repositories
Setup Tool and Layerindex

- Discovery and dependency management
- Manages additional buildtools
- Integrated as `bitbake-layers layerindex-fetch`
- Setup can also integrate layers not in layerindex
Developer Builds

also known as try builds
Developer Builds

- Developer build area with local commits
- Repo detects commits
- "Fork" repos and store commits on Git server
- Containerized Layerindex uses forks
- Recreate local build area on build system
Triggering Builds

- What configurations? How many builds?
- Default set of configs and builds
- Select machine, image, distro, sdk
- `local.conf` override
- Live feedback and email summary
- For all options see: `start_devbuild.sh`
Build Failure Login

For those nasty build failures
Build Failure Login

- Save Docker Container Startup command
- Create temp ssh key and upload to build server
- Add authorized keys entry to run docker on login
- Build environment perfectly recreated
- Cleanup keys
- See \texttt{workspace\_login.sh}
Extensible Pipeline

- OEQ tests on Qemu and Hardware using LAVA
- Test Failure Login
- Add more stages
Summary

- Expose large compute cluster to developers
- Build and test patches before commit
- Build failure inspection
- Integration with runtime testing
Future Work

- Yocto autobuilder helper scripts
- Using AWS to scale builds
- K8s and Tekton
- Yocto self-hosted AMIs
THANK YOU!!