Getting Started With New KernelCI CLI Tools

Automating Linux Kernel Testing and Validation

Your Name
KernelCI / Gentoo / etc.

Motivation: Why Kernel QA Is Hard

- Many trees, branches and configurations
- Multiple architectures, boards and toolchains
- CI dashboards are powerful but not ergonomic
- Developers still click UIs or write ad-hoc scripts
- Command-line tools keep context in the terminal

KernelCI in One Slide

- Upstream, open testing for the Linux kernel
- Builds, boots and tests across distributed labs
- Great data, but interaction is:
  - Mostly via web UI
  - Raw REST APIs
- Need something ergonomic in the terminal

Real-World Context: gentoo-sources

- Fast-moving tree with many patches
- High pressure to avoid regressions
- Need quick, scriptable visibility into CI results
- KernelCI CLI tools are built with this reality in mind

Introducing the KernelCI CLI Tools

- **kci-dev**
  - Developer-focused CLI for interacting with KernelCI
- **kci-deploy** *(work in progress)*
  - Tool for deploying local / internal KernelCI maestro stacks
- Goal: make KernelCI a first-class tool in your terminal

kci-dev: What You Can Do Today

Examples (adapt to real syntax):

```
kci-dev builds list --branch topic/foo
kci-dev results summary --commit <sha>
kci-dev builds retry --build-id <id>
```

kci-dev: Everyday Workflow

1. **Discover builds**
   - Filter by branch, commit, maintainer or config
2. **Inspect results quickly**
   - Summaries by status, platform, architecture
3. **Retrigger or bisect**
   - Kick a rebuild or boot test when needed
4. **Download artifacts**
   - Logs, dmesg, reports for offline inspection

kci-dev: Quality-of-Life Details

- Rich output modes: table, JSON, and quiet for scripts
- Persistent auth + endpoint profiles
- Smart defaults (branch from git, latest results)
- Designed for piping into jq, fzf, or notebooks

kci-deploy: For Lab Owners

- Simplifies standing up a Maestro stack
- Encodes best practices for networking and storage
- Same CLI UX as kci-dev
- Early previews welcome: help shape the roadmap!

Automation Patterns

- **Pre-submit checks**: gate merges on KernelCI signal
- **Nightly reports**: email/Matrix summaries via cron
- **Release readiness**: track blockers for RCs
- **Local sanity tests**: run focused boards before shipping

Demo Script (Idea)

```
# Configure once
kci-dev profile add ossj --url https://linux.kernelci.org --token $TOKEN

# Morning health check
kci-dev results summary --branch main --since 2d

# Investigate a failure
kci-dev logs show --job <id> | less

# Nudge infra
kci-dev builds retry --build-id <id>
```

Roadmap & Collaboration

- Deeper git integration (auto-pick branch/commit)
- Better diffing between runs
- Inline links back to dashboards
- kci-deploy installer previews in Q2
- Looking for testers, lab partners, and feedback

Getting Started After This Talk

- Try **kci-dev** with your tree this week
- Share your top 3 pain points in Kernel QA
- Join the KernelCI community calls / Matrix
- Contribute docs, plugins, and issue repros

Thank You!

Slides: https://example.com

@yourhandle

Questions welcome