

The background features a light-colored, stylized illustration of a city skyline. On the left, a large, curved pipe with a blue vertical section and a brown horizontal section extends from the top. Below the pipe is a small penguin icon. The city skyline consists of various buildings, including a large cathedral with multiple spires and a central archway. The sky is a pale yellow with soft, horizontal clouds. The water in the foreground is a light blue with some darker blue patches.

Relevance of LTS distros and GenAI world

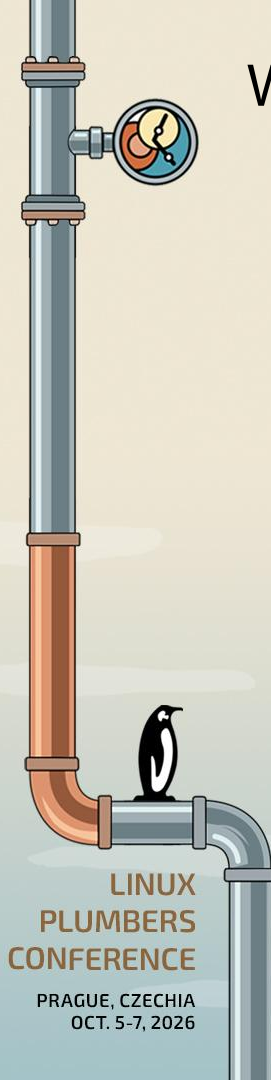


LINUX
PLUMBERS
CONFERENCE

PRAGUE, CZECHIA / OCT. 5-7, 2026

Why LTS Distributions Still Matter

- A known kernel, libc, compiler, linker, package-manager, and host-tool baseline
- A stable qualification target for CI and product validation
- Coordinated security maintenance
- Predictable package and ABI policies
- Reduced variability between development, CI, and production
- A practical base for regulated and long-lived devices
- A common recovery and field-update environment



LINUX
PLUMBERS
CONFERENCE

PRAGUE, CZECHIA
OCT. 5-7, 2026

Current Release Models Are Already a Spectrum

Fixed stable/LTS

- | Stable plus backports
- | Stable base plus moving layers
- | Periodic snapshots
- | Continuously qualified mainline



Rolling release

LINUX
PLUMBERS
CONFERENCE

PRAGUE, CZECHIA
OCT. 5-7, 2026

GenAI Changes Patch Production

Before coding agents

Issue —► Maintainer analysis —► Patch —► Review —► CI

With coding agents

CVEs

Version updates

Build failures

API changes

CI failures

Metadata cleanup

Upstream changes

► Agents —► Many candidate patches

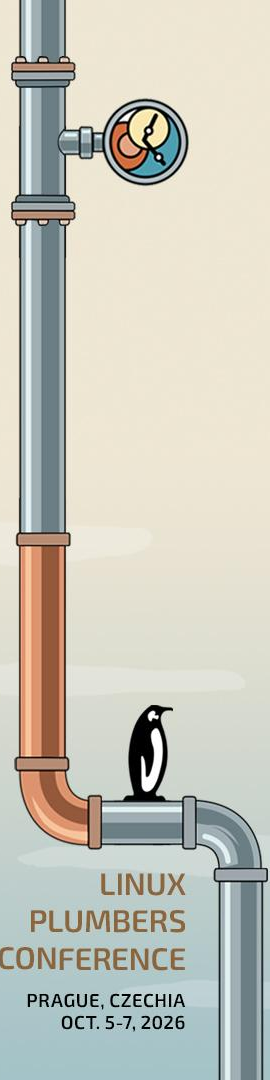
|



Review + validation

LINUX
PLUMBERS
CONFERENCE

PRAGUE, CZECHIA
OCT. 5-7, 2026

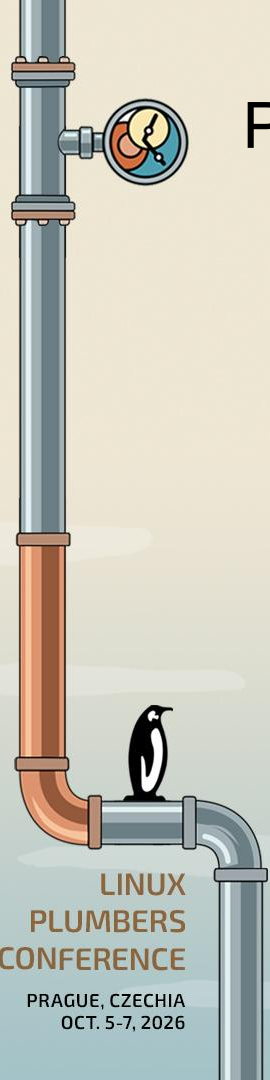


Coding agents increase the production capacity of contributors, but they do not automatically increase the verification capacity of maintainers.

Will coding agents remove maintenance work, or merely move the bottleneck downstream?

LINUX
PLUMBERS
CONFERENCE

PRAGUE, CZECHIA
OCT. 5-7, 2026



Patch Volume Is Not Maintenance Capacity

- A syntactically correct patch may not be a semantically correct update.
- A successful build does not prove functional compatibility.
- A package upgrade may change transitive dependencies.
- A backport may not receive the same upstream test coverage as the original fix.
- Embedded projects need testing across architectures, machines, toolchains, and product configurations.
- The submitter may be an agent, but the responsibility remains human and communal.
- More patches can produce more abandoned patches, review queues, and false confidence.

LINUX
PLUMBERS
CONFERENCE

PRAGUE, CZECHIA
OCT. 5-7, 2026

LTS May Become More Expensive

Upstream velocity

×

Number of supported LTS branches

×

Architectures and configurations

×

Dependency combinations

×

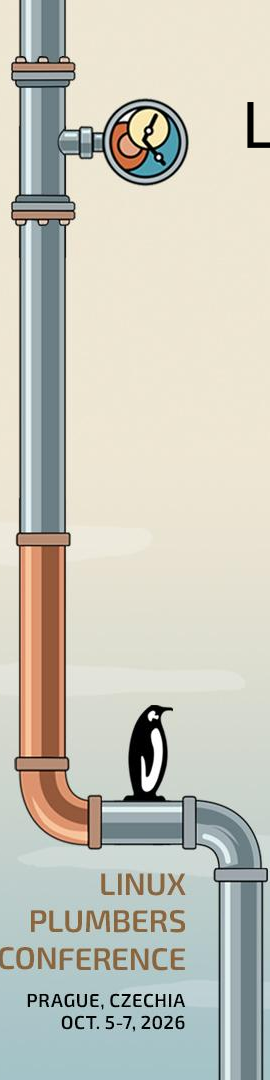
Validation requirements

=

Maintenance and CI burden

LINUX
PLUMBERS
CONFERENCE

PRAGUE, CZECHIA
OCT. 5-7, 2026



LTS May Become More Expensive

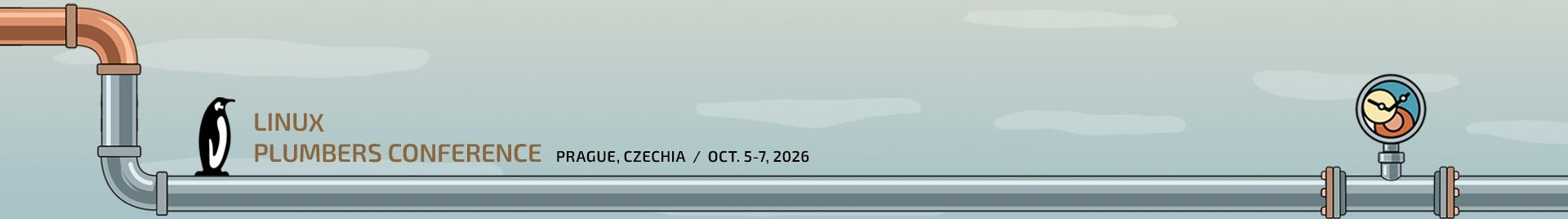
GenAI may increase the arrival rate of candidate changes faster than projects can expand review and test capacity.

At what point does an LTS branch become a continuously rebased downstream fork rather than a stable release?

If GenAI can help maintain an LTS branch, should we spend that capacity preserving old branches or improving mainline?

Is Rolling Release Viable - Advantages

- Less long-term backporting
- Faster access to new hardware and compiler support
- Smaller divergence from upstream
- Security fixes arrive through the current development stream
- Greater alignment with what upstream maintainers test
- AI-agent effort can target current code



Is Rolling Release Viable - Challenges

- Product certification and qualification
- Long hardware lifetimes
- Expensive field testing
- Interdependent firmware, bootloader, kernel, and userspace changes
- Proprietary components with slower update cadences
- Offline or constrained update environments
- Customer expectations of stable interfaces
- Need for reliable rollback and recovery
- Difficulty reproducing a historical rolling state without precise manifests



LINUX

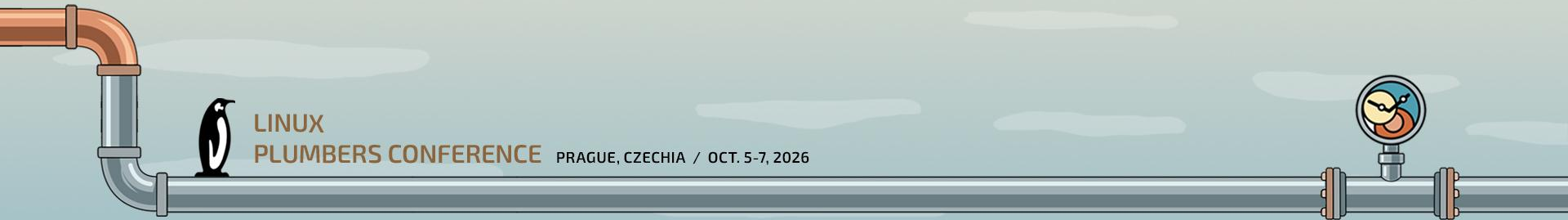
PLUMBERS CONFERENCE

PRAGUE, CZECHIA / OCT. 5-7, 2026



Continuously Qualified mainline

- Mainline remains the primary development branch.
- Automation and agents focus on current upstream code.
- Broad CI continuously measures release readiness.



Something to ponder upon

Where should genAI effort go?

How should conventional LTS evolve?

Who remains accountable?

When generating a patch becomes cheap, how should open-source projects allocate the scarce resources of review, validation, and long-term responsibility?

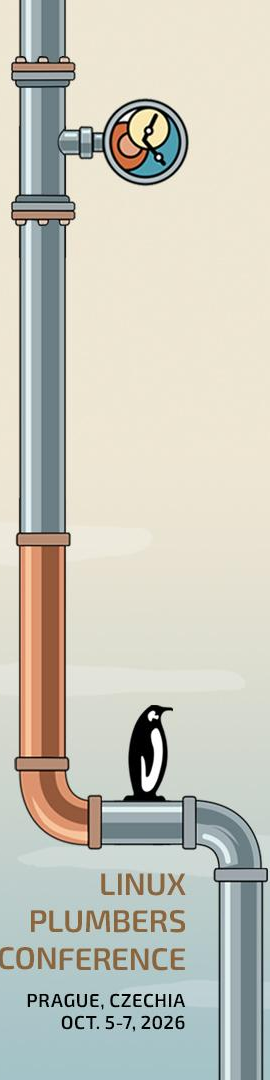


LINUX

PLUMBERS CONFERENCE

PRAGUE, CZECHIA / OCT. 5-7, 2026



A vertical pipe with a pressure gauge and a penguin on a horizontal section. The pipe is blue with orange joints. A pressure gauge is attached to the upper part. A penguin is perched on a horizontal section of the pipe. The background is a light blue gradient with white clouds.

Thanks for your time

LINUX
PLUMBERS
CONFERENCE

PRAGUE, CZECHIA
OCT. 5-7, 2026