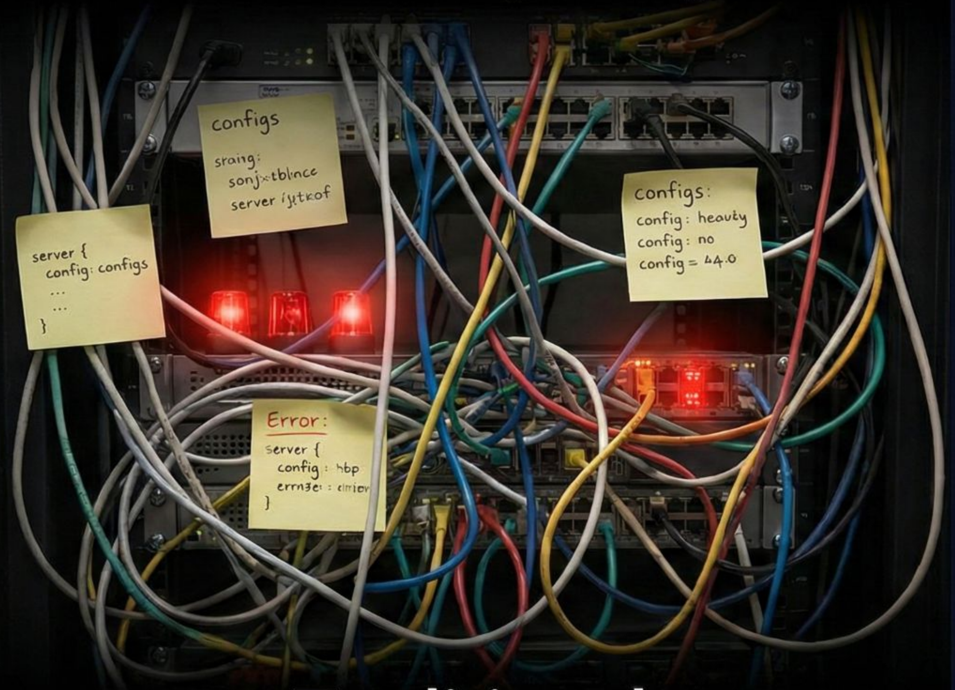
NixOS for Deterministic Distributed-System Benchmarking

Fosdem ‘26

Bruce Gain
Analyst | ReveCom
bruce@revecom.io

Reproducible Benchmarking at Scale

Eliminating Configuration Drift with **NixOS** & Cassandra



Traditional

Nix

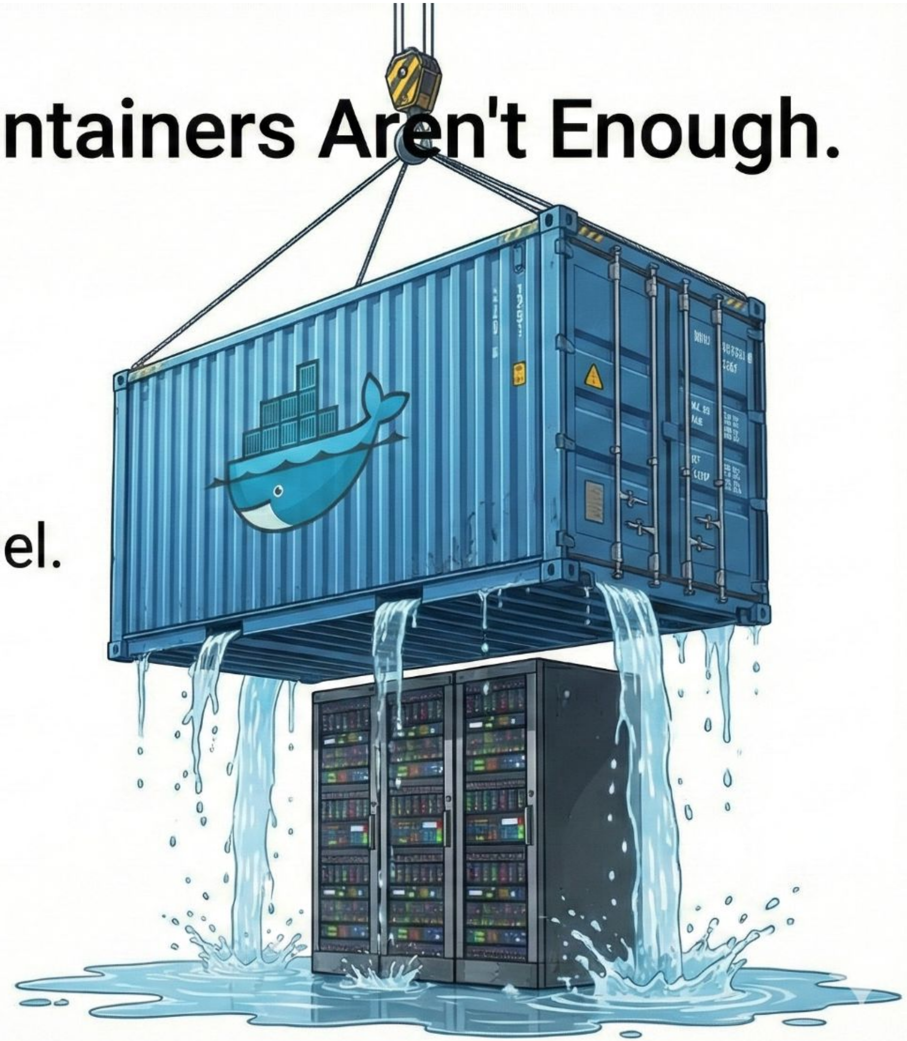
The Villain: Why Benchmarks Lie.

- Change only one variable at a time.
- The Reality: Configuration Drift.
- Silent kernel updates, different library versions.
- The Cost: Measuring the noise, not the database.

The False Hope: Why Containers Aren't Enough.

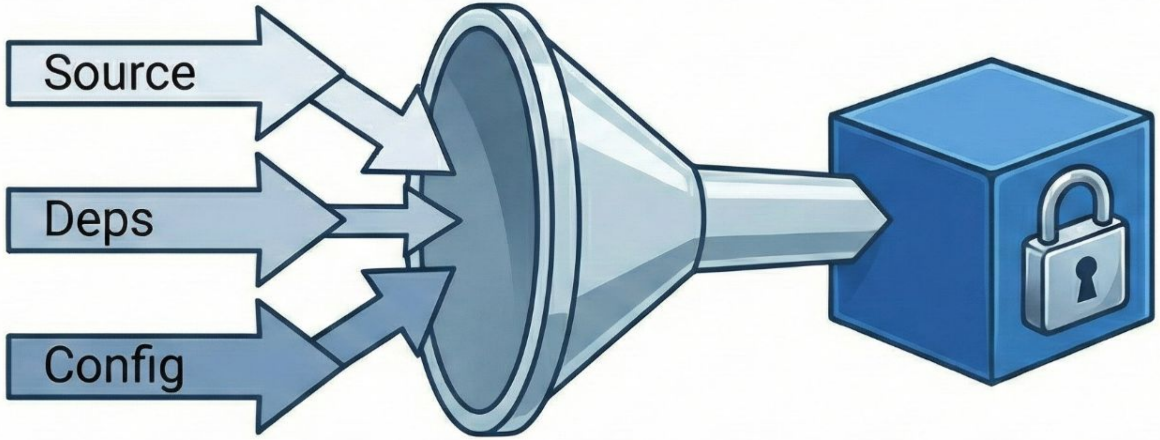
- Docker solved packaging, not total reproducibility.
- The "Leaky" Abstraction: Containers share the host kernel.
- Different host kernel =

  different results.
- The Build Problem: `apt-get update` is not reproducible.



The Hero: Determinism through Inputs.

- Treating Infrastructure as a Pure Function.
- If inputs are identical, output *must* be identical.



Inputs (Source + Dependencies + Config) = Output Hash

The Mechanism: The Power of the Hash.

sah762k...9a8b7c

xyp123z...4d5e6f

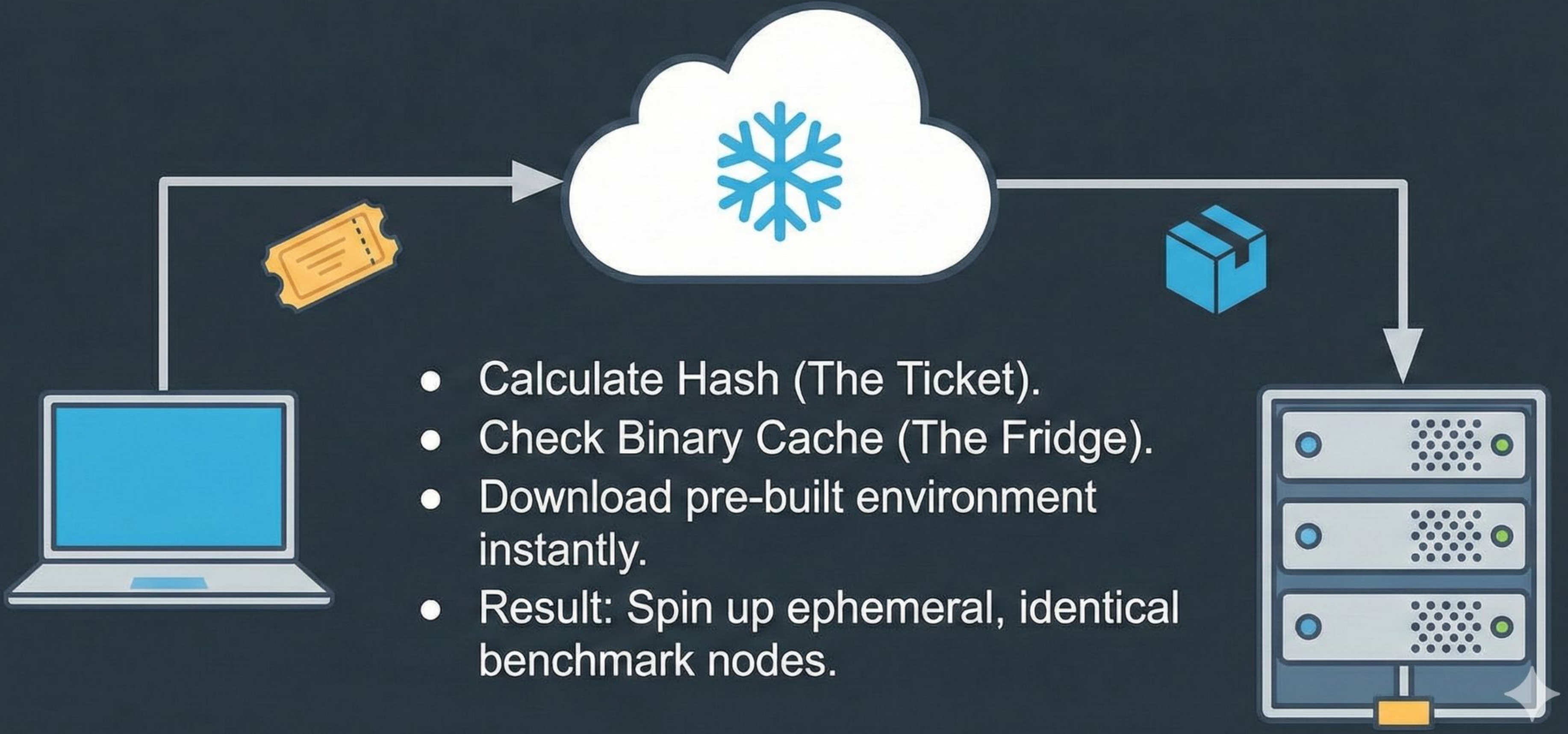
Change one character... get a completely NEW hash.

- Unique cryptographic fingerprint for every component.
- It’s not an “upgrade.” It is a replacement.

Case Study: Taming Apache Cassandra.

- Cassandra is sensitive to *everything* (JVM, Kernel, Disk I/O).
- The Nix Approach: Define all variables as immutable inputs.

The Workflow: Build Once, Run Anywhere.



- Calculate Hash (The Ticket).
- Check Binary Cache (The Fridge).
- Download pre-built environment instantly.
- Result: Spin up ephemeral, identical benchmark nodes.

Thank you! Questions? Q&A

Cassandra on Nix package
github.com/ReveCom/nix-talk-demo

Acknowledgements

Nix   NixOS   Cassandra   Shahid Khahn

Contact Info

bruce@revecom.io   www.revecom.io

ReveCom MEDIA