Twitter Thread by Kristian Köhntopp



Kristian Köhntopp
@isotopp



Unpopular opinion: CentOS switching to Stream is not a bad thing at all.

When you wanted to run Openstack on CentOS in 2015, you needed to enable https://t.co/paegQ99rAj to even begin an install. The first thing that did was literally replace every single package in the install.

That was, because CentOS at that time was literally making Debian Stale look young.

And we see similar problems with Ubuntu LTS, fwiw. LTS comes out every 2 years, and that's kind of ok-ish, but it lasts 5 years, which is nonsensical.

It was not, in the past. So what changed?

Software Development.

We have been moving to a platform based development approach, leveraging the wins from DevOps.

"Kris, that's corporate bullshit."

It's not, though. Let me spell it out in plain for you.

People these days do not program in an editor, with a compiler.

They use Github or Gitlab, with many integrations, and a local IDE. They commit to a VCS (the VCS, it's spelled git), and trigger a bunch of things.

Typechecks, reformatters, tests, but also code quality metrics and security scanners.

Starting a new programming language in 2020 is not as easy as it was in the past, because you do not only need a language, but also a JetBrains product, SonarQube support, gitlab-ci.yaml's…

We have come to rely on an entire ecosystem of tooling to make our developers faster, and to enforce uniform standards across the group. And that is a good thing.

We also have come to rely on tooling to enable collaboration, and structured discussion about code.

At the production end of that platform, we have made our base operating systems largely immutable - just enough infrastructure to run docker or k8s, and set up the actual production code in movable and removable units, containers.

Building these units is automated, and produces reproducible artifacts as part of a CD pipeline and repositories. We can roll forward and backwards, provided we manage state appropriately.

This - containers and VMs, also immutable, and runtime config, also versioned and injected using appropriate mechanisms - made everything single user and single tenant.

Side Note: sudo-curl-bash doesn't matter as much as everybody make it to be, because inside your container there is likely only one user, anyway.

We use other, better mechanisms that UIDs for separation these days.

That in turn enables us to architect systems from building blocks.

So your Wifi Control Plane system uses three Java processes and one Mongodb. Docker it into one container, or provide a helm chart or docker-compose with four for better scalability and maintenance.

So your gitlab deploy uses a Postgres, a Redis, and a dozen other components. Fine, deploy it as an Omnibus container or a K8s deploy yaml, and now you get a dozen components running.

So Jitsi is a Prosody, three to five Java components and even more stuff? Fine, deploy a single docker-compose.yaml and it assembles itself almost automatically.

On almost any Linux distro as a substrate.

Exactly. The actual distro does hardly matter at all, because your software comes packaged with all dependencies anyway, or is even built locally that way.

This allows us to move on.

Independently. On a per-project basis.

And the project will bring its own database, cache, runtime and libraries with itself, without version conflicts, and without waiting for the distro to move or to provide at all.

Distros moving to Stream is just the consequence of that.

They are finally free from slow moving OSS projects preventing them from upgrading local components because they can't keep up.

Even teams in Enterprises are now free to move at their own speed, because they no longer have to wait for half a dozen stakeholders to get to the Technical Debt Section of their backlog.

CentOS moving to Streams is not only acknowledging that change, it also forced the slower half of the world to acknowledge this, and embrace it.

I say "Yay, finally, progress." ■✔■■

And if you rant "Stability goes out of the window!" - check your calendar and your processes.

It's 2020. Act like it.

Der Completeness wegen noch die Originalquelle:

https://t.co/FwQhepoO0V

"Wenn eure Compliance darauf basiert das laufende Endprodukt zu zertifizieren, wird euere Organisation nicht mit der Entwicklung mithalten können." -- @hikhvar

https://t.co/fm8IATTrg7
This is now a blog article.