Chiselled Ubuntu containers

Valentin Viennot – Canonical
September 14, 2022
How and why (Ubuntu) distroless?

Provenance + Security concerns with Open Source Software

- Containerisation is not enough to secure an application
- Abstracting dependencies also abstracts vulnerabilities
- Layering mechanism ⇒ many containers based on the same content
- Keeping content up-to-date with containers isn’t straightforward
How and why (Ubuntu) distroless?
How and why (Ubuntu) distroless?

- There's a correlation between size of the image and number of CVEs
How and why (Ubuntu) distroless?

- There’s a correlation between size of the image and number of CVEs
- Reducing the size helps but isn’t enough
- Content provenance matters!
- And Developer Experience + ecosystem + support also matters

+ reducing the size benefits both small and at-scale environments
  (reduces storage and memory resources consumption)
How and why (Ubuntu) distroless?

Containers didn’t kill Linux distributions....

... but ...

... there’s a security and resources consumption challenge to solve
How and why (Ubuntu) distroless?


** (+) **
- No package manager
- No shell
- About 20MB
- Useful as/for runtime images

** (-) **
- Complex to use
- Complex to build
- No support
- Built with Bazel
How and why (Ubuntu) distroless?


(+)
No package manager
No shell
About 20MB
Useful as/for runtime images

(-)
Complex to use
Complex to build
No support
Built with Bazel
Could we have the advantages of a Linux distribution

... without the overhead?
How and why (Ubuntu) distroless?

- Chiselled Ubuntu containers for <insert-app-or-runtime>
Chiselled Ubuntu containers?

- <12MB for an “Ubuntu Distroless base”
- No package manager (avoid whole class of attacks)
- No shell (avoid whole class of attacks)
- Based on known and supported Ubuntu packages
- Compatible developer experience from host/server/container/chiselled

  ○ *from (70MB, build)*
  
  FROM ubuntu:22.04
  
  *to (13MB, run)*
  
  FROM ubuntu/dotnet-deps:22.04