Nix en DevOps: entornos consistentes y reproducibles sin esfuerzo

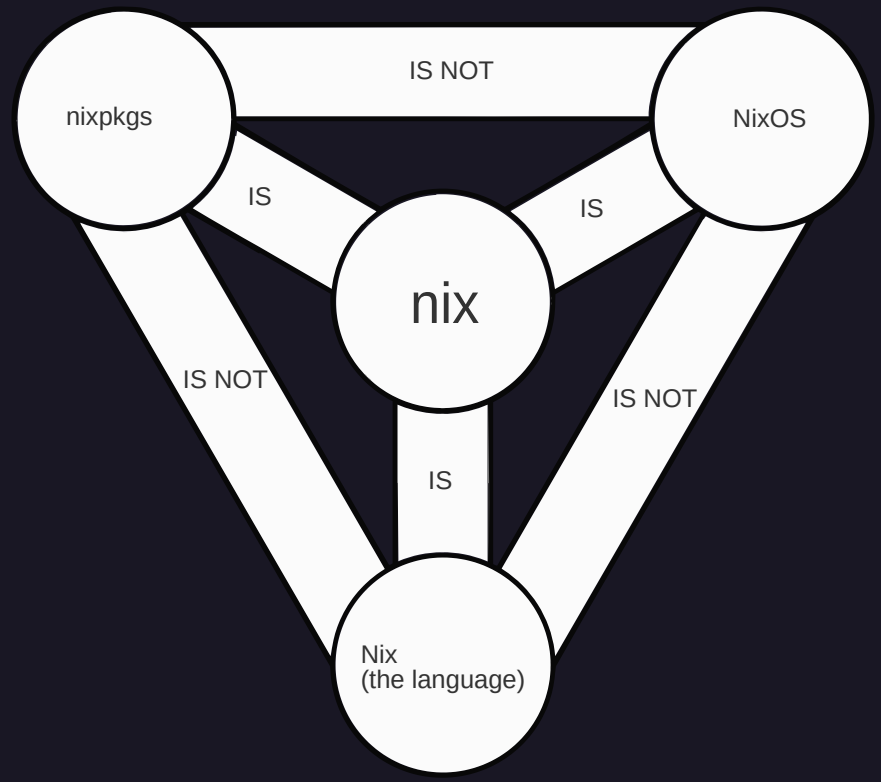
Valencia DevOps, 20/11/2024

$ Whoami

- Apasionado del software libre
- Estudiante de doctorado en la Universidad de Castilla-La Mancha
- Miembro del Summer of Nix 2024
- Fediverse: @amerinor01@mastodon.social
- Matrix: @amerino:matrix.org

¿Que es Nix?

- Gestor de Paquetes
- Sistema Operativo
- Lenguaje de Programación
- Ecosistema integral diseñado para desarrollar, implementar y gestionar software de forma reproducible.



Orígenes de Nix

- Bases académicas -> Solucionar los principales problemas de la construción del software
  - Reproducibilidad:"*Build once, run everywhere*"
  - Inmutabilidad: Garantiza estados idempotentes
  - Declarativo
- Dolstra, E. (2006). The purely functional software deployment model.

Por que importa a los DevOps

- Aliniación con los principios de CI/CD
- Ya no existe el caso de "Funciona en mi maquina"
- Permite una mejores entornos compartidos entre los equipos de Dev y DevOps

Reproducibilidad

- Todo en nix parte de una derivación
- Las derivaciones contienen información:
  - Dependencias
  - instrucciones de compilación / instalación
  - versiones y metadatos de la derivación
- Cada Dependencia es una propia derivación

Reproductibilidad

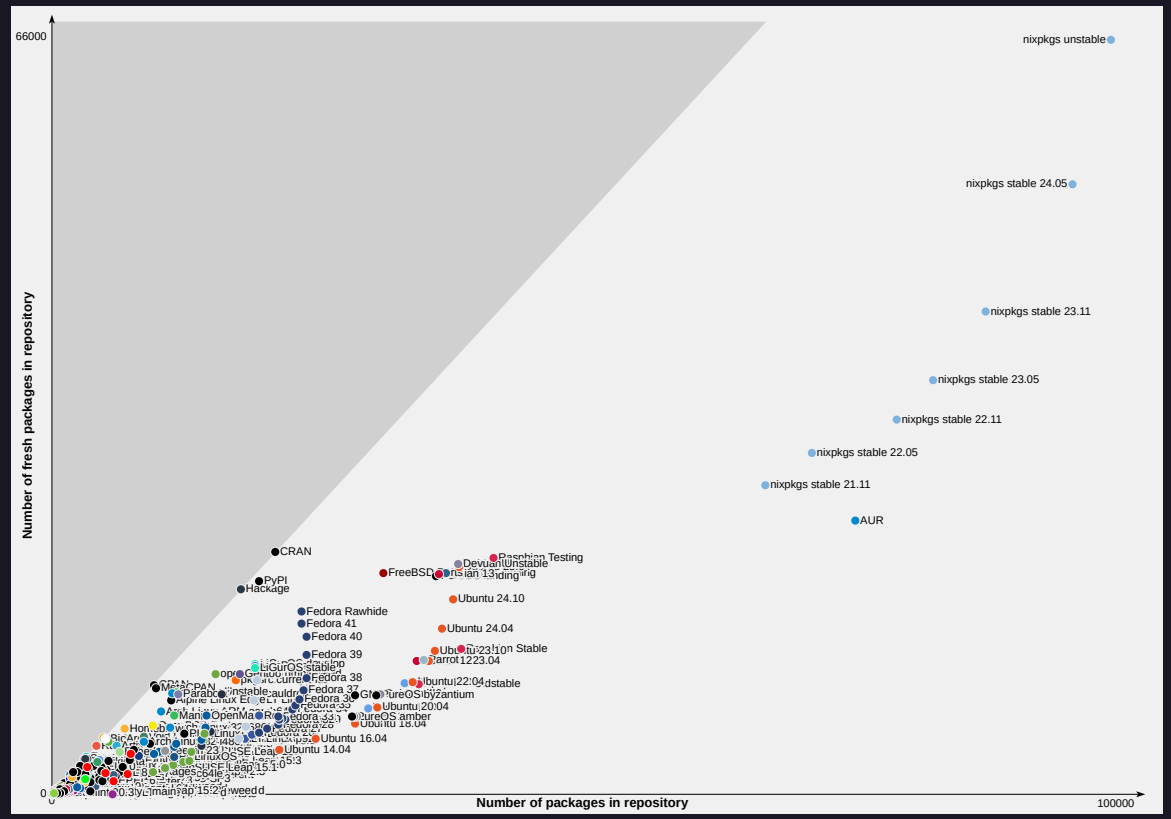
- Cada derivación se introduce en `/nix/store/...` con un nombre único en función del resultado del binario/biblioteca
  - Multiples versiones pueden coexistir sin generar conflictos.
- Bibliotecas aisladas para cada derivación.

Reproductibilidad

- **Aislamiento de red**: Nix bloquea el acceso a internet durante la construcción de paquetes para evitar dependencias no declaradas.
    - Esto garantiza que las construcciones sean reproducibles y no dependan del estado externo.

Nixpkgs

- Mas de 100.000 paquetes en Nixpkgs
- Compatibilidad multiplataforma
- Permite el uso de caches para acelerar las actualizaciones



NixOS

- Configuración total del sistema con expresiones Nix
- **Actualizaciones seguras y reversibles:** Las actualizaciones son atómicas, y puedes revertir fácilmente a la configuración anterior
- Sistema Operativo declarativo e inmutable.
- Soporte para entornos heterogéneos
  - Perminte la configuración de múltiples dispositivos con diferentes configuraciones (HPC, IoT, Escritorios)

Herramientas para CI/CD

- Hydra: Construcción de pipelines de CI con Nix
- Cachix: Herramienta de caches para derivaciones Nix

Estandarización del entorno de trabajo con Nix

- DevBox: Configura rápidamente espacios de trabajo aislados para pruebas y desarrollo.
- DevShell: Crea entornos de desarrollo bajo demanda, adaptados a las necesidades del proyecto.
- Flakes: Un estándar moderno para la reproducibilidad y la gestión estructurada de proyectos
    - Incluido en Nix.

Casos de Uso Avanzados

- NixOps: Automatización de despliegues
- Construcción y ejecución de contenedores con dockerTools
- Colmena: Orquestación ligera
- Home Manager: Entornos de usuario declarativos
- Integraciones con herramientas como Terraform, Helm(con Kubenix), etc.

Demo Time

Retos y limitaciones

Adopción en la Comunidad

- Tecnología de nicho, pero en auge.
- La comunidad es muy activa, pero aún hay áreas que mejorar, comola documentación

Flakes

- Características experimentales.
- Todavía no están estandarizados.

Nix en DevOps: entornos consistentes y reproducibles sin esfuerzo

Valencia DevOps, 20/11/2024