NIXOS

Malte Neuss

NIXOS

“Reproducible builds and deployments.”

By Tim Cuthbertson CC BY 4.0, https://nixos.org

CONTENT

Nix

- Package Manager
- Language
- OS

NIX PACKAGE MANAGER

- Adhoc environments

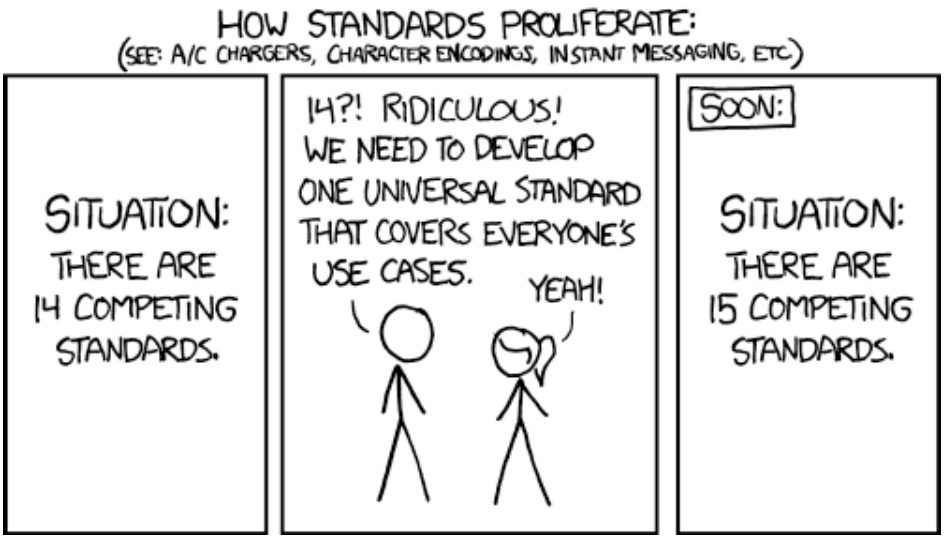
```
$ nix-shell --packages jdk
```

- Reproducible environments

```
mkDerivation {
  name = "myProject-1.0.0";
  src = ./src;
  buildInputs = [ maven jdk ...];
}
```

DEMO

NIX LANGUAGE



By https://xkcd.com/927/ CC BY-NC 2.5

NIX LANGUAGE

Derivation

```
{
  "/nix/store/dvmidxj5...-myProject-1.0.0.drv": {
    "outputs": {
      "out": {
        "path": "/nix/store/w9yy7v...-myProject-1.0.0"
      }
    },
    "inputSrcs": [
      "/nix/store/9krlzvn...-default-builder.sh"
    ],
    "inputDrvs": {
      "/nix/store/1psqjc0l1..-bash-4.4-p23.drv": [
        "out"
      ],
```

NIX LANGUAGE

≈Json + Functions

```
mkDerivation {
  name = "myProject-1.0.0";
  src = ./src;
  buildInputs = [ maven jdk ];
}
```

Benefits

- Declarative
- Functional
- Pure

NIX LANGUAGE

Pure, Functional, Lazy

- Strings `"hello"`
- Numbers 42
- List `[1 2 3]`
- Expressions 1+2+3
- Attribute Sets `{ }`
- Functions `f x`

ATTRIBUTE SET

```
{
  name = "myProject-1.0.0";
  src = ./src;
  buildInputs = [ maven jdk ];
  ...
}
```

Assignment:

```
myNumber   = 3;
myAttrSet  = { name = "myProject-1.0.0"; src = ... }
myString   = myAttrSet.name;
myFunction = x: 2*x;
```

FUNCTIONS

Typescript

```
const f = x => 2x
f(3)
```

Nix

```
f = x: 2*x          # definition
f 3                 # usage
```

FUNCTIONS

Typescript

```
const g = (x,y) => x+y
g(3,4)
```

Nix

```
g = x: y: x+y
g 3 4
```

FUNCTIONS

Currying

```
g: (ℝ,ℝ) → ℝ
g: (x,y) ↦ x+y
```

```
g: ℝ → (ℝ → ℝ)
g: x ↦ (y ↦ x+y)
```

Usage

```
g (3,4)
```

```
g 3 4
```

FUNCTIONS

```
mkDerivation {
  name = "myProject-1.0.0";
  src = ./src;
  ..
}
```

Resulting Derivation


```
{
  "/nix/store/dvmig8jgrdapvbyxb1rprckdmdqx08kv-myProject-
        1.0.0.drv": {
    "outputs": {
      "out": {
        "path": "/nix/store/w9yy7v61ipb5rx6i35zq1mvc2iqfmps1-
        myProject-1.0.0"
      }
    },
    "inputSrcs": [
      "/nix/store/9krlzvny65gdc8s7kpb6lkx8cd02c25b-default-
        builder.sh"
    ],
    "inputDrvs": {
```

DERIVATION

```
# shell.nix
let
 pkgs = import <nixpkgs> {};
in
pkgs.stdenv.mkDerivation {
  name = "myProject-1.0.0";
  src = ./src;
  buildInputs = [ pkgs.maven pkgs.jdk ];
  buildPhase = "maven clean build";
}
```

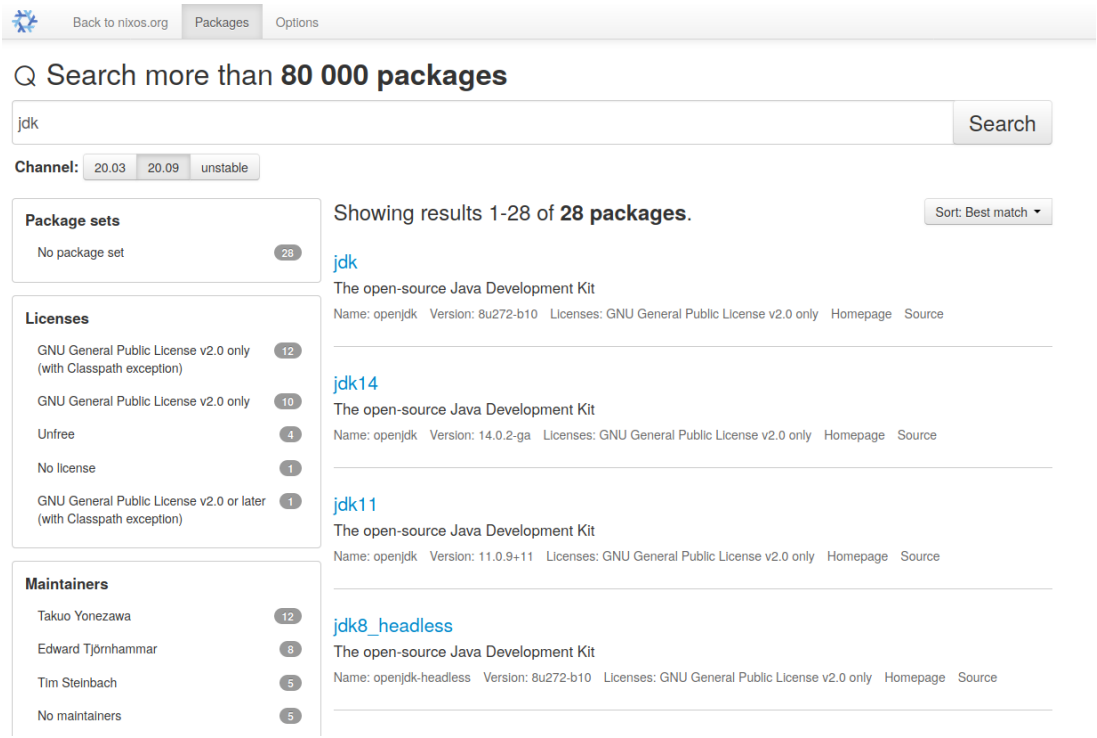
NIXPKGS

(import <nixpkgs> {}) evaluates to:

```
{
  stdenv = { .. }
  ...
  maven = stdenv.mkDerivation {... jdk ...}
  jdk = ..
  jdk11 = ..
  ...
  git = ....
  ...
  firefox = ...
  ...
}
```

https://github.com/NixOS/nixpkgs

SEARCH PACKAGES



https://search.nixos.org

REPRODUCIBLE BUILDS

```
$ nix-build shell.nix   # build artifact, put into store
```

Store

```
/nix/store/9234jfkdfj23j45r2102jfd-myProject-1.0.0
/nix/store/34234sdfjskdfj32j4kjdsf-maven-3.0.0
/nix/store/sdf34dfkjlkj09u123123ss-maven-3.0.0
..
/nix/store/fsdkf234jdfdsfjkj0111df-jdk-1.8.0
..
```

```
$ nix-shell shell.nix    # load environment with buildInputs
```

DEMO

NIXOS

```
{
  ..
  boot.loader.grub.device = "/dev/sda";
  boot.kernelPackages = pkgs.linuxPackages_latest;
  ..

  users.users.mneuss = {
    isNormalUser = true;
    home = "/home/mneuss";
    extraGroups = [
      "wheel" # Enable ‘sudo’ for the user.
      "docker"
      "network-manager" ];
```

```
nixos-rebuild switch
```

FURTHER STUDY

- **Main Page**
  https://nixos.org/
- **Cloud Deployment**
  https://github.com
  /NixOS/nixops
- **My Blog**
  https://lambdablob.com/
- **Switch Dev Environments**
  https://direnv.net/

// reveal.js plugins