The Nix Package Manager

Eelco Dolstra
e.dolstra@tudelft.nl

Delft University of Technology, EWI,
Department of Software Technology

November 12, 2009

Nix

- Nix: purely functional package manager
- NixOS: Linux distribution based on Nix
- Hydra: continuous build system based on Nix
- `http://nixos.org/`

What's wrong with other package managers?

- Upgrading a package is dangerous
- Hard to have multiple versions of a package installed at the same time
- Upgrades are not atomic
- No rollbacks
- Incomplete dependency info
- Only root can install packages
- ...

Nix: Purely functional package management

Nix is a purely functional package manager.

- Purely functional language to describe how to build packages and their dependencies
- Build results only depend on declared inputs.
- Packages never change after they have been built.

Nix store

Main idea: store all packages in isolation from each other:

`/nix/store/rpdqxnilb0cg...-firefox-3.5.4`

Paths contain a 160-bit **cryptographic hash** of **all** inputs used to build the package:

- Sources
- Libraries
- Compilers
- Build scripts
- ...

```
/nix/store
├─ l9w6773m1msy...-openssh-4.6p1
│  ├─ bin
│  │  └─ ssh
│  └─ sbin
│     └─ sshd
├─ smkabrbibqv7...-openssl-0.9.8e
│  └─ lib
│     └─ libssl.so.0.9.8
├─ c6jbqm2mc0a7...-zlib-1.2.3
│  └─ lib
│     └─ libz.so.1.2.3
└─ im276akmsrhv...-glibc-2.5
   └─ lib
      └─ libc.so.6
```

Nix expressions

openssh.nix

```
{ stdenv, fetchurl, openssl, zlib }:

stdenv.mkDerivation {
  name = "openssh-4.6p1";
  src = fetchurl {
    url = http://.../openssh-4.6p1.tar.gz;
    sha256 = "0fpjlr3bfind0y94bk442x2p...";
  };
  buildCommand = ''
    tar xjf $src
    ./configure --prefix=$out --with-openssl=${openssl}
    make; make install
  '';
}
```

Nix expressions

all-packages.nix

```
openssh = import ../tools/networking/openssh {
  inherit fetchurl stdenv openssl zlib;
};

openssl = import ../development/libraries/openssl {
  inherit fetchurl stdenv perl;
};

stdenv = ...;
openssl = ...;
zlib = ...;
perl = ...;
}
```

Nix expressions

all-packages.nix

```
openssh = import ../tools/networking/openssh {
  inherit fetchurl stdenv openssl zlib;
};

openssl = import
  inherit fetchu
};

stdenv = ...;
openssl = ...;
zlib = ...;
perl = ...;
}
```

Evaluating the openssh variable will produce an OpenSSH package in the Nix store.

```
/nix/store
├─ l9w6773m1msy...-openssh-4.6p1
│  ├─ bin
│  │  └─ ssh
│  └─ sbin
│     └─ sshd
└─ ...
```

User operations

- To build and install OpenSSH:

```
$ nix-env -f all-packages.nix -i openssh
```

- When a new version comes along:

```
$ nix-env -f all-packages.nix -u openssh
```

- If it doesn't work:

```
$ nix-env --rollback
```

- Delete unused components:

```
$ nix-collect-garbage
```

User operations

- To build and install OpenSSH:

```
$ nix-env -f all-packages.nix -i openssh
```

- When a new version comes along:

```
$ nix-env -f all-packages.nix -u openssh
```

- If it doesn't work:

```
$ nix-env --rollback
```

- Delete unused components:

```
$ nix-collect-garbage
```

User operations

- To build and install OpenSSH:

```
$ nix-env -f all-packages.nix -i openssh
```

- When a new version comes along:

```
$ nix-env -f all-packages.nix -u openssh
```

- If it doesn't work:

```
$ nix-env --rollback
```

- Delete unused components:

```
$ nix-collect-garbage
```

User operations

- To build and install OpenSSH:

```
$ nix-env -f all-packages.nix -i openssh
```

- When a new version comes along:

```
$ nix-env -f all-packages.nix -u openssh
```

- If it doesn't work:

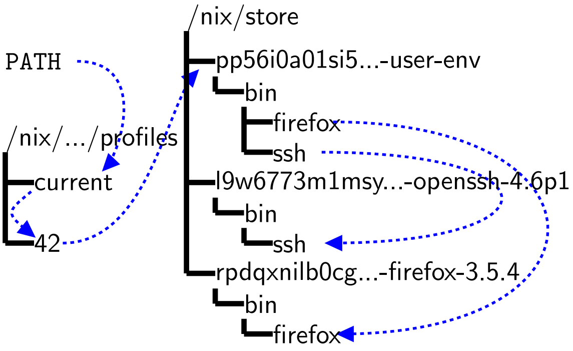
```
$ nix-env --rollback
```

- Delete unused components:

```
$ nix-collect-garbage
```

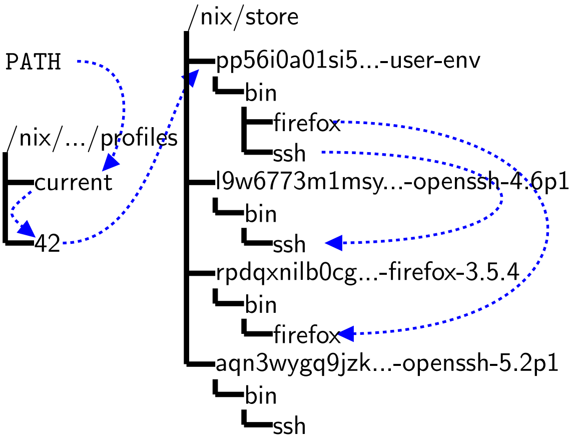
User environments

- Users can have different sets of installed applications.
- `nix-env` operations create new **user environments** in the store.
- We can atomically switch between them.
- These are roots of the **garbage collector**.



User environments

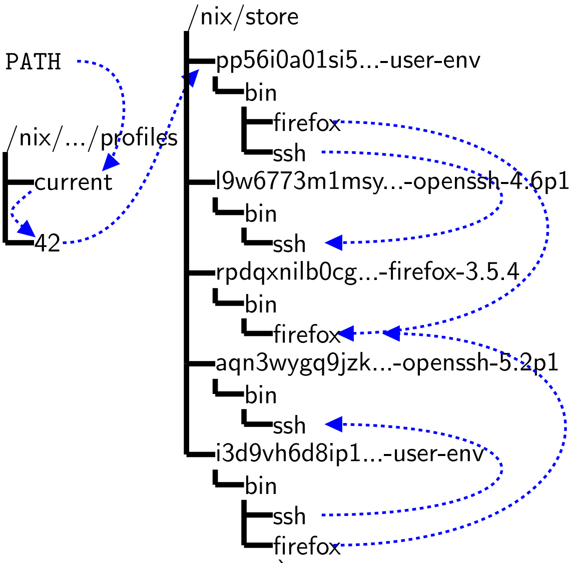
- Users can have different sets of installed applications.
- `nix-env` operations create new **user environments** in the store.
- We can atomically switch between them.
- These are roots of the **garbage collector**.



(`nix-env -u openssh`)

User environments

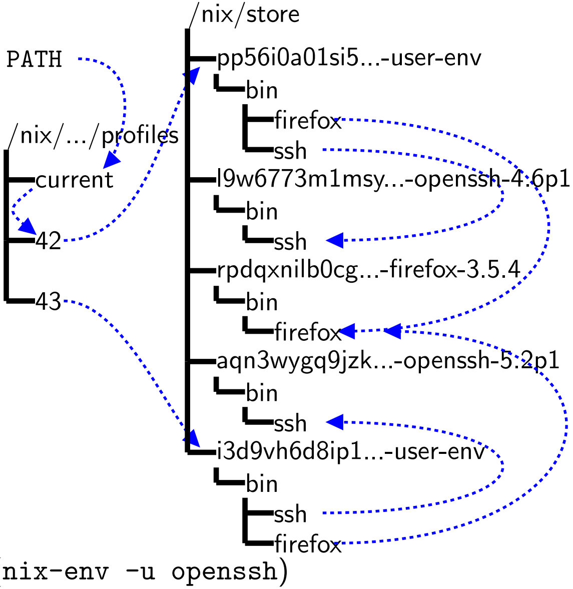
- Users can have different sets of installed applications.
- `nix-env` operations create new **user environments** in the store.
- We can atomically switch between them.
- These are roots of the **garbage collector**.



(`nix-env -u openssh`)

User environments

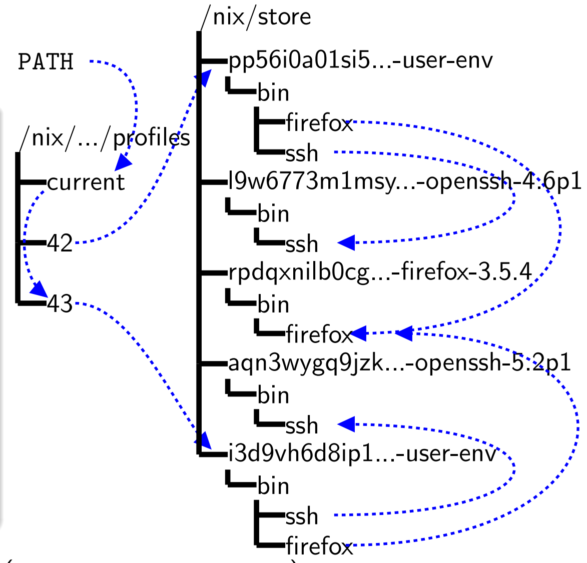
- Users can have different sets of installed applications.
- `nix-env` operations create new **user environments** in the store.
- We can atomically switch between them.
- These are roots of the **garbage collector**.



(`nix-env -u openssh`)

User environments

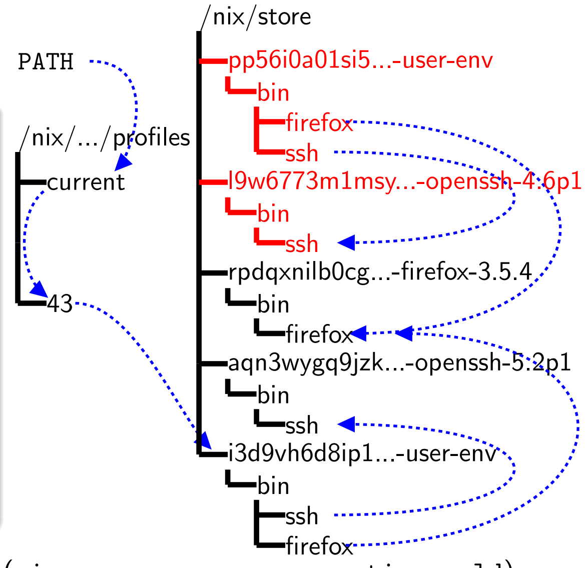
- Users can have different sets of installed applications.
- `nix-env` operations create new **user environments** in the store.
- We can atomically switch between them.
- These are roots of the **garbage collector**.



(`nix-env -u openssh`)

User environments

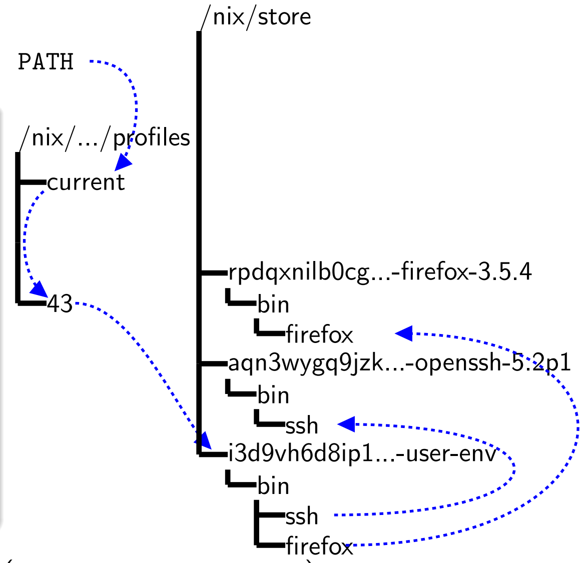
- Users can have different sets of installed applications.
- `nix-env` operations create new **user environments** in the store.
- We can atomically switch between them.
- These are roots of the **garbage collector**.



(`nix-env --remove-generations old`)

User environments

- Users can have different sets of installed applications.
- `nix-env` operations create new **user environments** in the store.
- We can atomically switch between them.
- These are roots of the **garbage collector**.



(`nix-collect-garbage`)

Deployment using Nix

- This is a **source deployment model** (like Gentoo), but...
- We get **binary deployment** by sharing pre-built components.
- On the producer side:

```
$ nix-push $(nix-instantiate all-packages.nix) \
   http://server/cache
```

- On the client side:

```
$ nix-pull http://server/cache
$ nix-env -f all-packages.nix -i openssh
```

- Installation will now reuse pre-built components, **iff** they are exactly the same.

Deployment using Nix

- This is a **source deployment model** (like Gentoo), but...
- We get **binary deployment** by sharing pre-built components.
- On the producer side:

```
$ nix-push $(nix-instantiate all-packages.nix) \
   http://server/cache
```

- On the client side:

```
$ nix-pull http://server/cache
$ nix-env -f all-packages.nix -i openssh
```

- Installation will now reuse pre-built components, **iff** they are exactly the same.

Deployment using Nix

- This is a **source deployment model** (like Gentoo), but...
- We get **binary deployment** by sharing pre-built components.
- On the producer side:

```
$ nix-push $(nix-instantiate all-packages.nix) \
   http://server/cache
```

- On the client side:

```
$ nix-pull http://server/cache
$ nix-env -f all-packages.nix -i openssh
```

- Installation will now reuse pre-built components, **iff** they are exactly the same.

Finding runtime dependencies

```
/nix/store
├─ l9w6773m1msy...-openssh-4.6p1
│  ├─ bin
│  │  └─ ssh
│  └─ sbin
│     └─ sshd
├─ smkabrbibqv7...-openssl-0.9.8e
│  └─ lib
│     └─ libssl.so.0.9.8
├─ c6jbqm2mc0a7...-zlib-1.2.3
│  └─ lib
│     └─ libz.so.1.2.3
└─ im276akmsrhv...-glibc-2.5
   └─ lib
      └─ libc.so.6
```

Finding runtime dependencies

```
/nix/store
├─ l9w6773m1msy...-openssh-4.6p1
│  ├─ bin
│  │  └─ ssh
│  └─ sbin
│     └─ sshd
├─ smkabrbibqv7...-
│  └─ lib
│     └─ libssl.so.0
├─ c6jbqm2mc0a7...-
│  └─ lib
│     └─ libz.so.1.2.3
└─ im276akmsrhv...-glibc-2.5
   └─ lib
      └─ libc.so.6
```

Contents of l9w6...-openssh-4.6p1/bin/ssh

```
...
72 74 00 5f 65 6e 64 00  2f 6e 69 78 2f 73 74 6f  |rt._end./nix/sto|
72 65 2f 35 6d 6a 30 35  31 30 66 78 6a 76 32 71  |re/c6jbqm2mc0a7q|
33 79 71 6c 71 76 79 72  70 68 37 37 34 69 79 6e  |3yqlqvyrph774iyn|
6b 6c 66 2d 7a 6c 69 62  2d 31 2e 32 2e 33 2f 6c  |klf-zlib-1.2.3/l|
69 62 3a 2f 6e 69 78 2f  73 74 6f 72 65 2f 32 6b  |ib:/nix/store/sm|
38 76 6a 6a 37 31 64 68  6d 38 73 72 33 67 6b 79  |kabrbibqv7sr3gky|
68 7a 33 64 67 7a 31 37  33 76 35 78 6b 67 2d 6f  |hz3dgz173v5xkg-o|
70 65 6e 73 73 6c 2d 30  2e 39 2e 38 6b 2f 6c 69  |penssl-0.9.8e/li|
...
```

Finding runtime dependencies

```
/nix/store
├─ l9w6773m1msy...-openssh-4.6p1
│  ├─ bin
│  │  └─ ssh
│  └─ sbin
│     └─ sshd
├─ smkabrbibqv7...-
│  └─ lib
│     └─ libssl.so.0
├─ c6jbqm2mc0a7...-
│  └─ lib
│     └─ libz.so.1.2.3
└─ im276akmsrhv...-glibc-2.5
   └─ lib
      └─ libc.so.6
```

Contents of l9w6...-openssh-4.6p1/bin/ssh

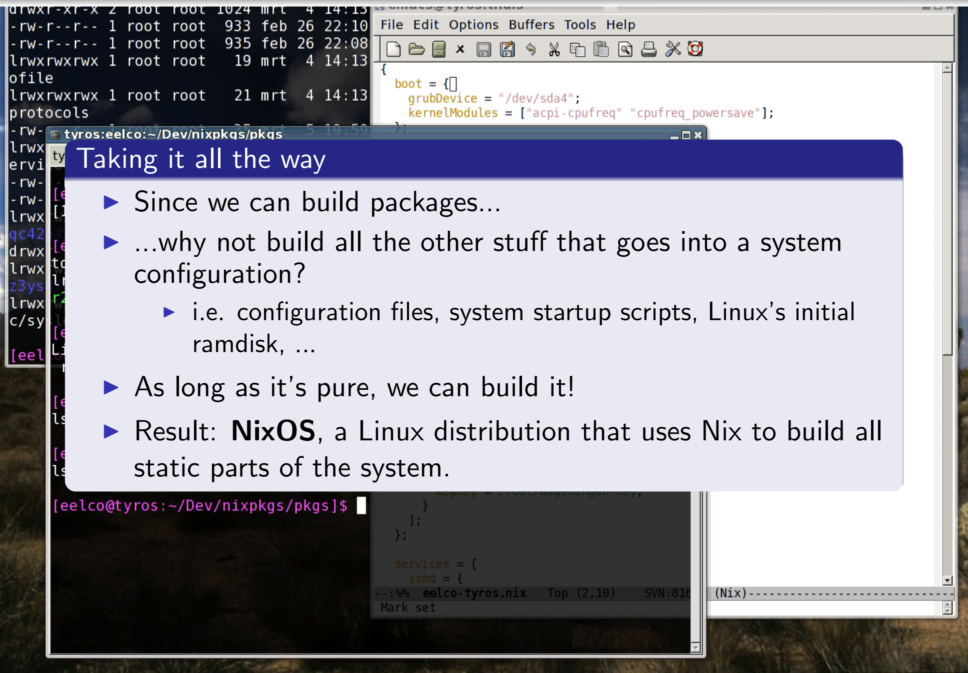
```
...
72 74 00 5f 65 6e 64 00  2f 6e 69 78 2f 73 74 6f  |rt._end./nix/sto|
72 65 2f 35 6d 6a 30 35  31 30 66 78 6a 76 32 71  |re/c6jbqm2mc0a7q|
33 79 71 6c 71 76 79 72  70 68 37 37 34 69 79 6e  |3yqlqvyrph774iyn|
6b 6c 66 2d 7a 6c 69 62  2d 31 2e 32 2e 33 2f 6c  |klf-zlib-1.2.3/l|
69 62 3a 2f 6e 69 78 2f  73 74 6f 72 65 2f 32 6b  |ib:/nix/store/sm|
38 76 6a 6a 37 31 64 68  6d 38 73 72 33 67 6b 79  |kabrbibqv7sr3gky|
68 7a 33 64 67 7a 31 37  33 76 35 78 6b 67 2d 6f  |hz3dgz173v5xkg-o|
70 65 6e 73 73 6c 2d 30  2e 39 2e 38 65 2f 6c 69  |penssl-0.9.8e/li|
...
```

Nix Packages collection

Nixpkgs

- Contains Nix expressions for $\geq$ 2100 existing Unix packages.
  - Development tools: GCC, Perl, Mono, ...
  - Libraries: Glibc, GTK, Qt, X11, ...
  - Applications: Firefox, OpenOffice, ...
  - Servers: Apache `httpd`, PostgreSQL, ...
- On Linux/x86, fully bootstrapped (no external dependencies).

NixOS


tyros:eelco:~/Dev/nixpkgs/pkgs
File Edit Options Buffers Tools Help

```
drwxr-xr-x 2 root root 1024 mrt  4 14:13
-rw-r--r-- 1 root root  933 feb 26 22:10
-rw-r--r-- 1 root root  935 feb 26 22:08
lrwxrwxrwx 1 root root   19 mrt  4 14:13
ofile
lrwxrwxrwx 1 root root   21 mrt  4 14:13
protocols
```

```
{
  boot = {
    grubDevice = "/dev/sda4";
    kernelModules = ["acpi-cpufreq" "cpufreq_powersave"];
  };
```

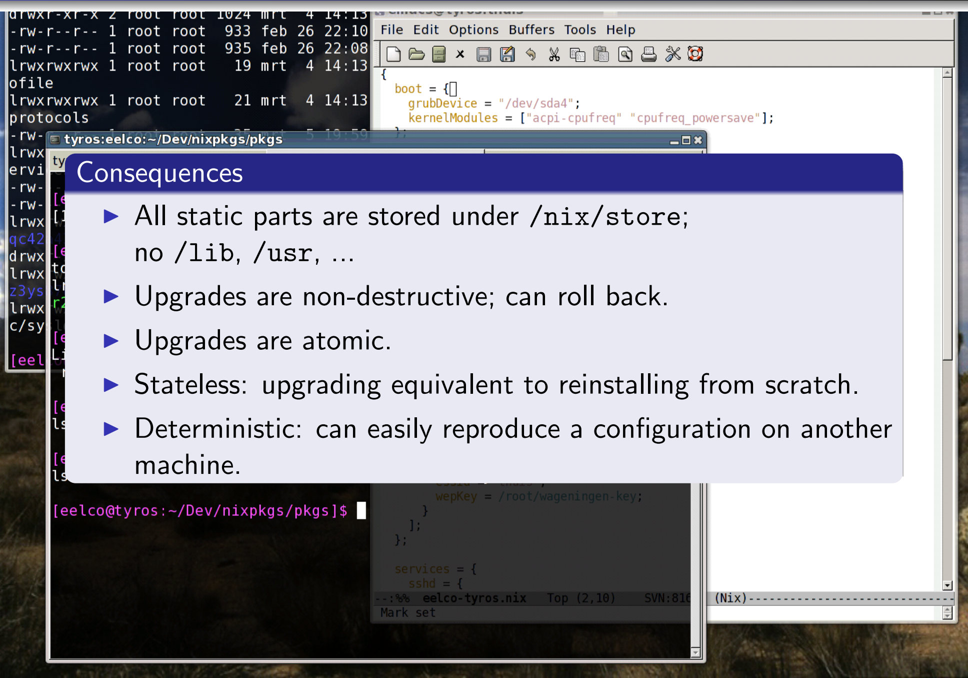
```
      wepkey = /root/wageningen-key;
    }
  ];
};

services = {
  sshd = {
--:%%  eelco-tyros.nix   Top (2,10)   SVN:816  (Nix)-----------------------------
Mark set
```

```
[eelco@tyros:~/Dev/nixpkgs/pkgs]$
```

Taking it all the way

- Since we can build packages...
- ...why not build all the other stuff that goes into a system configuration?
  - i.e. configuration files, system startup scripts, Linux's initial ramdisk, ...
- As long as it's pure, we can build it!
- Result: **NixOS**, a Linux distribution that uses Nix to build all static parts of the system.

NixOS


tyros:eelco:~/Dev/nixpkgs/pkgs
File Edit Options Buffers Tools Help

```
drwxr-xr-x 2 root root 1024 mrt  4 14:13
-rw-r--r-- 1 root root  933 feb 26 22:10
-rw-r--r-- 1 root root  935 feb 26 22:08
lrwxrwxrwx 1 root root   19 mrt  4 14:13
ofile
lrwxrwxrwx 1 root root   21 mrt  4 14:13
protocols
```

```
{
  boot = {
    grubDevice = "/dev/sda4";
    kernelModules = ["acpi-cpufreq" "cpufreq_powersave"];
```

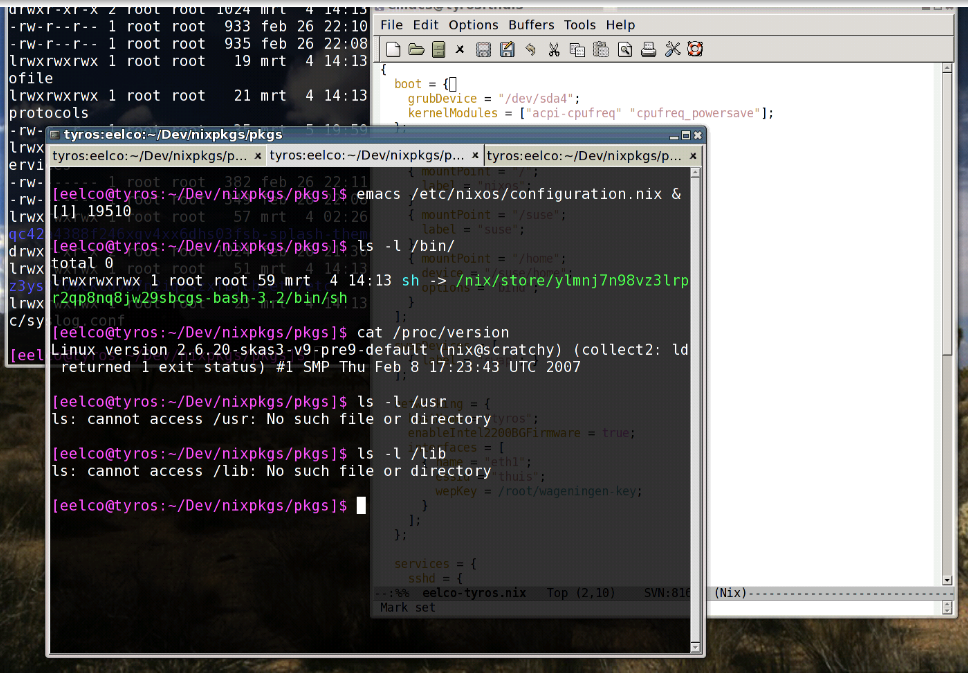
```
      wepKey = /root/wageningen-key;
    }
  ];
};

services = {
  sshd = {
--:%%  eelco-tyros.nix    Top (2,10)    SVN:816    (Nix)---------------------------
Mark set
```

```
[eelco@tyros:~/Dev/nixpkgs/pkgs]$
```

Consequences

- All static parts are stored under `/nix/store`; no `/lib`, `/usr`, ...
- Upgrades are non-destructive; can roll back.
- Upgrades are atomic.
- Stateless: upgrading equivalent to reinstalling from scratch.
- Deterministic: can easily reproduce a configuration on another machine.

NixOS

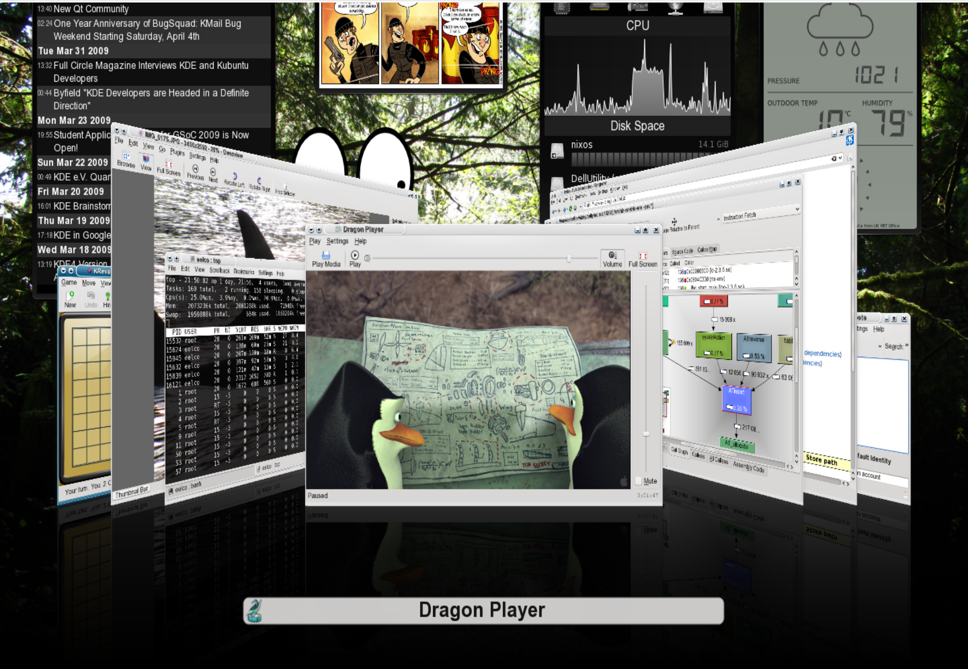


```
drwxr-xr-x 2 root root 1024 mrt  4 14:13
-rw-r--r-- 1 root root  933 feb 26 22:10
-rw-r--r-- 1 root root  935 feb 26 22:08
lrwxrwxrwx 1 root root   19 mrt  4 14:13
ofile
lrwxrwxrwx 1 root root   21 mrt  4 14:13
protocols
```

```
File  Edit  Options  Buffers  Tools  Help
{
  boot = {
    grubDevice = "/dev/sda4";
    kernelModules = ["acpi-cpufreq" "cpufreq_powersave"];
--:%%  eelco-tyros.nix    Top (2,10)    SVN:816    (Nix)----------------------------
Mark set
```

```
tyros:eelco:~/Dev/nixpkgs/pkgs
tyros:eelco:~/Dev/nixpkgs/p...  tyros:eelco:~/Dev/nixpkgs/p...  tyros:eelco:~/Dev/nixpkgs/p...

[eelco@tyros:~/Dev/nixpkgs/pkgs]$ emacs /etc/nixos/configuration.nix &
[1] 19510

[eelco@tyros:~/Dev/nixpkgs/pkgs]$ ls -l /bin/
total 0
lrwxrwxrwx 1 root root 59 mrt  4 14:13 sh -> /nix/store/ylmnj7n98vz3lrp
r2qp8nq8jw29sbcgs-bash-3.2/bin/sh

[eelco@tyros:~/Dev/nixpkgs/pkgs]$ cat /proc/version
Linux version 2.6.20-skas3-v9-pre9-default (nix@scratchy) (collect2: ld
 returned 1 exit status) #1 SMP Thu Feb 8 17:23:43 UTC 2007

[eelco@tyros:~/Dev/nixpkgs/pkgs]$ ls -l /usr
ls: cannot access /usr: No such file or directory

[eelco@tyros:~/Dev/nixpkgs/pkgs]$ ls -l /lib
ls: cannot access /lib: No such file or directory

[eelco@tyros:~/Dev/nixpkgs/pkgs]$ 
```

NixOS



13:40 New Qt Community
02:24 One Year Anniversary of BugSquad: KMail Bug Weekend Starting Saturday, April 4th
Tue Mar 31 2009
13:32 Full Circle Magazine Interviews KDE and Kubuntu Developers
00:44 Byfield "KDE Developers are Headed in a Definite Direction"
Mon Mar 23 2009
19:55 Student Applic... GSoC 2009 is Now Open!
Sun Mar 22 2009
00:49 KDE e.V. Quar...
Fri Mar 20 2009
16:01 KDE Brainstorm...
Thu Mar 19 2009
17:18 KDE in Google...
Wed Mar 18 2009
13:19 KDE4 Version...

```
top - 21:50:02 up 1 day, 21:56,  4 users,  load avera
Tasks: 168 total,   2 running, 158 sleeping,   0 stop
Cpu(s): 25.0%us,  3.9%sy,  0.0%ni, 70.9%id,  0.0%wa,
Mem:   2073236k total,  2001288k used,    71948k free
Swap:  1959088k total,      684k used,  1659204k free
```

Example

Nix expression for `ssh_config`

```
{ config, pkgs }:

pkgs.writeText "ssh_config" ''
  SendEnv LANG LC_ALL ...
  ${if config.services.sshd.forwardX11 then ''
    ForwardX11 yes
    XAuthLocation ${pkgs.xorg.xauth}/bin/xauth
  '' else ''
    ForwardX11 no
  ''}
''
```

Example

Nix expression for ssh_config

```
{ config, pkgs }:

pkgs.writeText "ssh_config" ''
  SendEnv LANG LC_ALL ...
  ${if config.services.sshd.forwardX11 then ''
    ForwardX11 yes
    XAuthLocation ${pkgs.xorg.xauth}/bin/xauth
  '' else ''
    Fo
  ''}
''
```

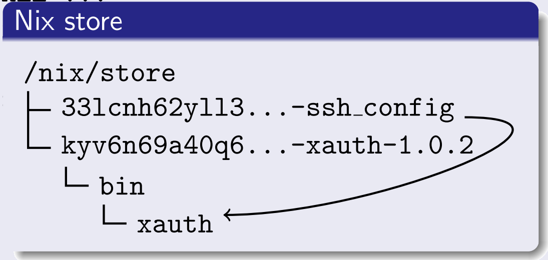
Laziness in action!

Example

Nix expression for `ssh_config`

```
{ config, pkgs }:

pkgs.writeText "ssh_config" ''
  SendEnv LANG LC_ALL ...
  ${if config.serv
    ForwardX11 yes
    XAuthLocation
  '' else ''
    ForwardX11 no
  ''}
''
```

Nix store



```
/nix/store
├─ 331cnh62yll3...-ssh_config
└─ kyv6n69a40q6...-xauth-1.0.2
   └─ bin
      └─ xauth
```

Example

Nix expression for ssh_config

```
{ config, pkgs }:

pkgs.writeText "ssh_config" ''
  SendEnv LANG LC_ALL ...
  ${if config.serv
    ForwardX11 yes
    XAuthLocation
  '' else ''
    ForwardX11 no
```

Nix store

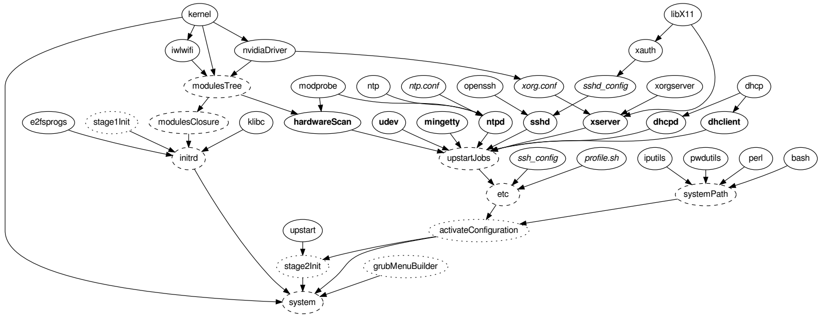
```
/nix/store
├─ 33lcnh62yll3...-ssh_config
└─ kyv6n69a40q6...-xauth-1.0.2
```

Generated file: 33lcnh62yll3...-sshd_config

```
SendEnv LANG LC_ALL ...
ForwardX11 yes
XAuthLocation /nix/store/kyv6n69a40q6...-xauth-1.0.2/bin/xauth
```

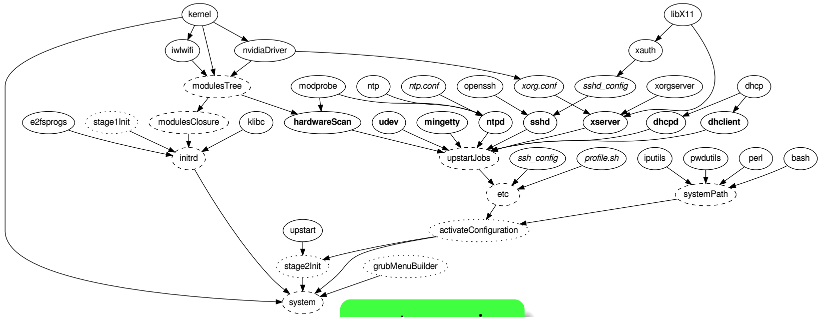
NixOS build time dependency graph

Nix expressions to build each part of the system: system packages, applications, their dependencies, kernel modules, initrd, configuration files, Upstart jobs, boot scripts, ...



NixOS build time dependency graph

Nix expressions to build each part of the system: system packages, applications, their dependencies, kernel modules, initrd, configuration files, Upstart jobs, boot scripts, ...



system.nix

The system configuration file

/etc/nixos/configuration.nix

```
{
  boot.loader.grub.bootDevice = "/dev/sda";
  fileSystems = singleton
    { mountPoint = "/";
      device = "/dev/sda1";
    };
  swapDevices = [ { device = "/dev/sdb1"; } ];
  services.sshd.enable = true;
  services.sshd.forwardX11 = true;
}
```

The system configuration file

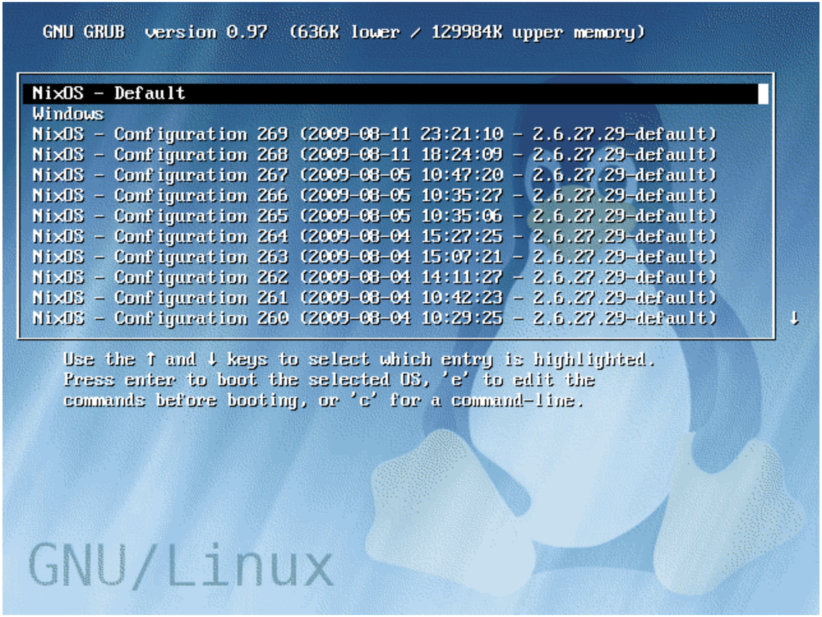
`/etc/nixos/configuration.nix`

```
{
  boot.loader.grub.bootDevice = "/dev/sda";
  fileSystems = singleton
    { mountPoint = "/";
      device = "/d
    };
  swapDevices = [
  services.sshd.en
  services.sshd.fo
}
```

End-user perspective

- Edit `configuration.nix`.
- Run `nixos-rebuild`.
- This builds `system.nix` and runs its activation script.
- Non-destructive; various rollback/test mechanisms.

NixOS — Grub boot menu

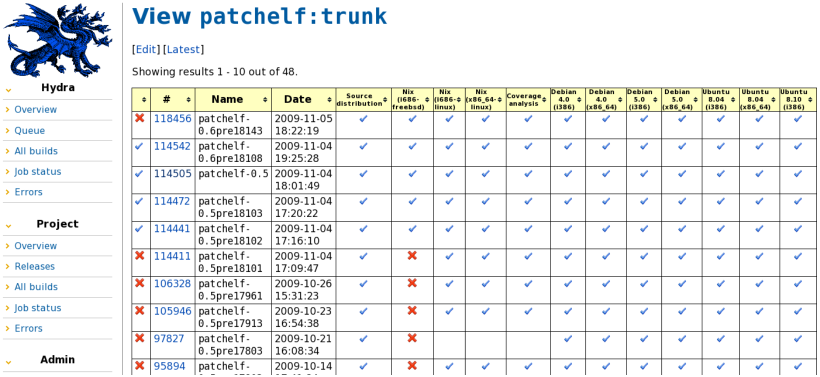


```
GNU GRUB  version 0.97  (636K lower / 129984K upper memory)

NixOS - Default
Windows
NixOS - Configuration 269 (2009-08-11 23:21:10 - 2.6.27.29-default)
NixOS - Configuration 268 (2009-08-11 18:24:09 - 2.6.27.29-default)
NixOS - Configuration 267 (2009-08-05 10:47:20 - 2.6.27.29-default)
NixOS - Configuration 266 (2009-08-05 10:35:27 - 2.6.27.29-default)
NixOS - Configuration 265 (2009-08-05 10:35:06 - 2.6.27.29-default)
NixOS - Configuration 264 (2009-08-04 15:27:25 - 2.6.27.29-default)
NixOS - Configuration 263 (2009-08-04 15:07:21 - 2.6.27.29-default)
NixOS - Configuration 262 (2009-08-04 14:11:27 - 2.6.27.29-default)
NixOS - Configuration 261 (2009-08-04 10:42:23 - 2.6.27.29-default)
NixOS - Configuration 260 (2009-08-04 10:29:25 - 2.6.27.29-default)  ↓

    Use the ↑ and ↓ keys to select which entry is highlighted.
    Press enter to boot the selected OS, 'e' to edit the
    commands before booting, or 'c' for a command-line.
```

Hydra

- Hydra: Continuous build system based on Nix
- Checks out projects from repos and builds them
- Build jobs described by Nix expressions
- Main advantage: builds all dependencies of a job

Hydra



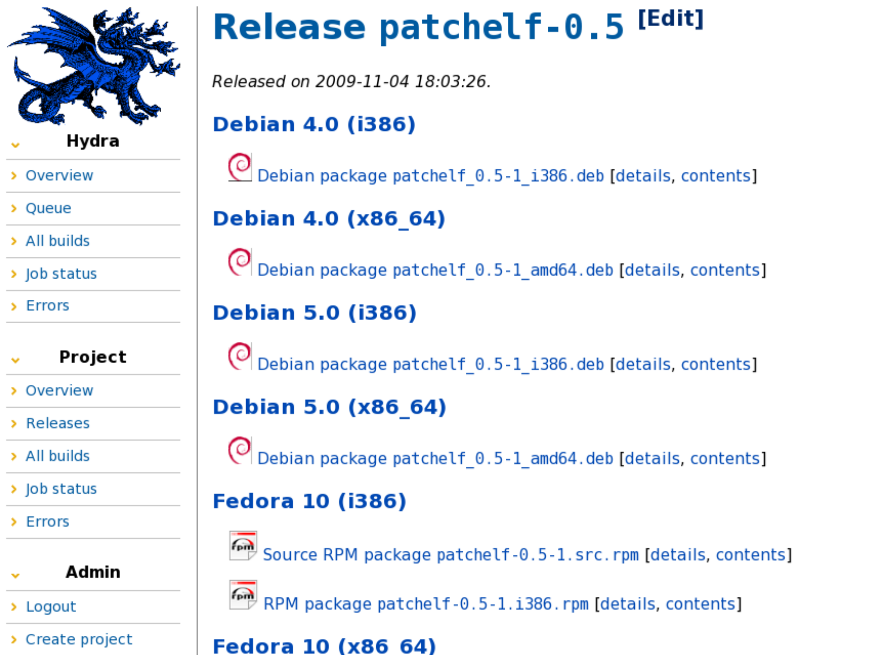
View patchelf:trunk

[Edit] [Latest]

Showing results 1 - 10 out of 48.

	#	Name	Date	Source distribution	Nix (i686-freebsd)	Nix (i686-linux)	Nix (x86_64-linux)	Coverage analysis	Debian 4.0 (i386)	Debian 4.0 (x86_64)	Debian 5.0 (i386)	Debian 5.0 (x86_64)	Ubuntu 8.04 (i386)	Ubuntu 8.04 (x86_64)	Ubuntu 8.10 (i386)
✗	118456	patchelf-0.6pre18143	2009-11-05 18:22:19	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
✓	114542	patchelf-0.6pre18108	2009-11-04 19:25:28	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
✓	114505	patchelf-0.5	2009-11-04 18:01:49	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
✓	114472	patchelf-0.5pre18103	2009-11-04 17:20:22	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
✓	114441	patchelf-0.5pre18102	2009-11-04 17:16:10	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
✗	114411	patchelf-0.5pre18101	2009-11-04 17:09:47	✓	✗	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
✗	106328	patchelf-0.5pre17961	2009-10-26 15:31:23	✓	✗	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
✗	105946	patchelf-0.5pre17913	2009-10-23 16:54:38	✓	✗	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
✗	97827	patchelf-0.5pre17803	2009-10-21 16:08:34	✓	✗				✓	✓	✓	✓	✓	✓	✓
✗	95894	patchelf-	2009-10-14	✓	✗	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓

Hydra



Hydra

- Overview
- Queue
- All builds
- Job status
- Errors

Project

- Overview
- Releases
- All builds
- Job status
- Errors

Admin

- Logout
- Create project

Release patchelf-0.5 [Edit]

Released on 2009-11-04 18:03:26.

Debian 4.0 (i386)

Debian package patchelf_0.5-1_i386.deb [details, contents]

Debian 4.0 (x86_64)

Debian package patchelf_0.5-1_amd64.deb [details, contents]

Debian 5.0 (i386)

Debian package patchelf_0.5-1_i386.deb [details, contents]

Debian 5.0 (x86_64)

Debian package patchelf_0.5-1_amd64.deb [details, contents]

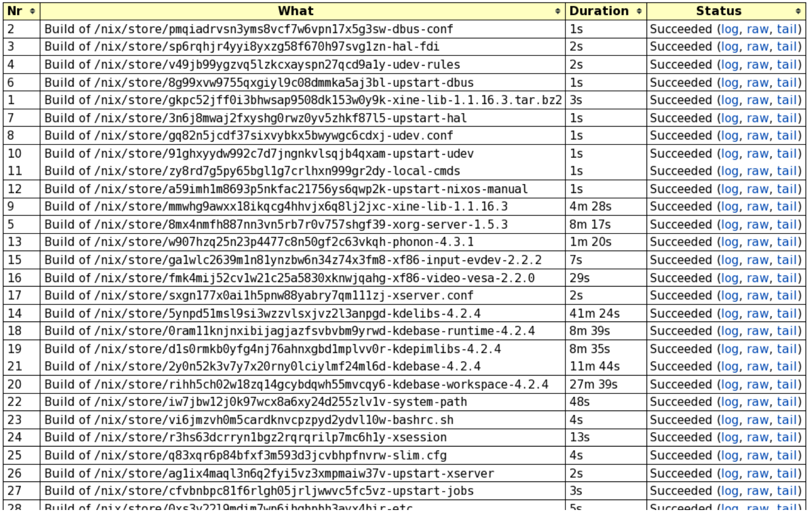
Fedora 10 (i386)

Source RPM package patchelf-0.5-1.src.rpm [details, contents]

RPM package patchelf-0.5-1.i386.rpm [details, contents]

Fedora 10 (x86_64)

Hydra



Nr	What	Duration	Status
2	Build of /nix/store/pmqiadrvsn3yms8vcf7w6vpn17x5g3sw-dbus-conf	1s	Succeeded (log, raw, tail)
3	Build of /nix/store/sp6rqhjr4yyi8yxzg58f670h97svg1zn-hal-fdi	2s	Succeeded (log, raw, tail)
4	Build of /nix/store/v49jb99ygzvq5lzkcxayspn27qcd9a1y-udev-rules	2s	Succeeded (log, raw, tail)
6	Build of /nix/store/8g99xvw9755qxgiyl9c08dmmka5aj3bl-upstart-dbus	1s	Succeeded (log, raw, tail)
1	Build of /nix/store/gkpc52jff0i3bhwsap9508dk153w0y9k-xine-lib-1.1.16.3.tar.bz2	3s	Succeeded (log, raw, tail)
7	Build of /nix/store/3n6j8mwaj2fxyshg0rwz0yv5zhkf87l5-upstart-hal	1s	Succeeded (log, raw, tail)
8	Build of /nix/store/gq82n5jcdf37sixvybkx5bwywgc6cdxj-udev.conf	1s	Succeeded (log, raw, tail)
10	Build of /nix/store/91ghxyydw992c7d7jngnkvlsqjb4qxam-upstart-udev	1s	Succeeded (log, raw, tail)
11	Build of /nix/store/zy8rd7g5py65bgl1g7crlhxn999gr2dy-local-cmds	1s	Succeeded (log, raw, tail)
12	Build of /nix/store/a59imh1m8693p5nkfac21756ys6qwp2k-upstart-nixos-manual	1s	Succeeded (log, raw, tail)
9	Build of /nix/store/mmwhg9awxx18ikqcg4hhvjx6q8lj2jxc-xine-lib-1.1.16.3	4m 28s	Succeeded (log, raw, tail)
5	Build of /nix/store/8mx4nmfh887nn3vn5rb7r0v757shgf39-xorg-server-1.5.3	8m 17s	Succeeded (log, raw, tail)
13	Build of /nix/store/w907hzq25n23p4477c8n50gf2c63vkqh-phonon-4.3.1	1m 20s	Succeeded (log, raw, tail)
15	Build of /nix/store/ga1wlc2639m1n81ynzbw6n34z74x3fm8-xf86-input-evdev-2.2.2	7s	Succeeded (log, raw, tail)
16	Build of /nix/store/fmk4mij52cv1w21c25a5830xknwjqahg-xf86-video-vesa-2.2.0	29s	Succeeded (log, raw, tail)
17	Build of /nix/store/sxgn177x0ai1h5pnw88yabry7qm111zj-xserver.conf	2s	Succeeded (log, raw, tail)
14	Build of /nix/store/5ynpd51msl9si3wzzvlsxjvz2l3anpgd-kdelibs-4.2.4	41m 24s	Succeeded (log, raw, tail)
18	Build of /nix/store/0ram11knjnxibijagjazfsvbvbm9yrwd-kdebase-runtime-4.2.4	8m 39s	Succeeded (log, raw, tail)
19	Build of /nix/store/d1s0rmkb0yfg4nj76ahnxgbd1mplvv0r-kdepimlibs-4.2.4	8m 35s	Succeeded (log, raw, tail)
21	Build of /nix/store/2y0n52k3v7y7x20rny0lciylmf24ml6d-kdebase-4.2.4	11m 44s	Succeeded (log, raw, tail)
20	Build of /nix/store/rihh5ch02w18zq14gcybdqwh55mvcqy6-kdebase-workspace-4.2.4	27m 39s	Succeeded (log, raw, tail)
22	Build of /nix/store/iw7jbw12j0k97wcx8a6xy24d255zlv1v-system-path	48s	Succeeded (log, raw, tail)
23	Build of /nix/store/vi6jmzvh0m5cardknvcpzpyd2ydvl10w-bashrc.sh	4s	Succeeded (log, raw, tail)
24	Build of /nix/store/r3hs63dcrryn1bgz2rqrqrilp7mc6h1y-xsession	13s	Succeeded (log, raw, tail)
25	Build of /nix/store/q83xqr6p84bfxf3m593d3jcvbhpfnvrw-slim.cfg	4s	Succeeded (log, raw, tail)
26	Build of /nix/store/ag1ix4maql3n6q2fyi5vz3xmpmaiw37v-upstart-xserver	2s	Succeeded (log, raw, tail)
27	Build of /nix/store/cfvbnbpc81f6rlgh05jrljwwvc5fc5vz-upstart-jobs	3s	Succeeded (log, raw, tail)
28	Build of /nix/store/0xs3v2219mdim7wp6ibqhpbh3ayx4hir-etc	5s	Succeeded (log, raw, tail)

Conclusion

- Nix: safe package management, atomic upgrades, rollbacks, multi-user, portable, ...
- NixOS: safe upgrades, atomic upgrades and rollbacks, reproducibility, ...
- Hydra: builds dependencies of a continuous build job automatically, ...

More information / download

- `http://nixos.org/`
- NixOS ISO images for x86, x86_64 are available.