nixpkgs-cookbook Documentation

Release 0.1

Domen Kožar

Jul 24, 2017

Contents

Source code GitHub

Bug tracker GitHub issues

Generated Jul 24, 2017

Version

Introduction

Nix Community Cookbook presents topical, practical ways of using Nix package manager ecosystem.

The Cookbook supplements the official Nix, NixOS, Nixkgs, Hydra and NixOps manuals.

To contribute to Nix Community Cookbook, please read *Contributing to the cookbook*.

CHAPTER 1

Table of Contents

Each Nix pill is intended to explain a certain part or concept in Nix or Nixpkgs.

Frequently Asked Questions

Nix

Secrets?

How do I fix: error: querying path in database: database disk image is malformed

Try:

> sqlite3 /nix/var/nix/db/db.sqlite “pragma integrity_check”

Which will print the errors in the database. If the errors are due to missing references, the following may work:

> mv /nix/var/nix/db/db.sqlite /nix/var/nix/db/db.sqlite-bkp sqlite3 /nix/var/nix/db/db.sqlite-bkp ”.dump” | sqlite3 /nix/var/nix/db/db.sqlite

How to operate between Nix paths and strings?

http://stackoverflow.com/a/43850372

How do I fix: error: current Nix store schema is version 10, but I only support 7

This means you have upgraded Nix sqlite schema to a newer version, but then tried to use older Nix.

The solution is to dump the db and use old Nix version to initialize it:

```
/path/to/nix/unstable/bin/nix-store --dump-db > /tmp/db.dump
mv /nix/var/nix/db /nix/var/nix/db.toonew
mkdir /nix/var/nix/db
nix-store --init (this is the old nix-store)
nix-store --load-db < /tmp/db.dump
```

How nix decides which parts of the environment affect a derivation and its sha256 hash

How to pin nixpkgs to a specific commit/branch?

Ways to provide/pin nixpkgs:

- `nix-channel --add URL nixpkgs && nix-channel --update` sets it globally for the user, but it doesn't allow precision (pinning to specific version)
- As environment variable: `$NIX_PATH=URL`
- `-I` command line parameter to most of commands like `nix-build`, `nix-shell`, etc
- Using builtins.fetchTarball function that fetches the channel at evaluation time

Possible `URL` values:

- Local file path. Using just `.` means that nixpkgs is located in current folder.
- Pinned to a specific commit: `https://github.com/NixOS/nixpkgs/archive/addcb0dddf2b7db505dae5c38fceb691c7ed85f9.tar.gz`
- Using latest channel, meaning all tests have passed: `http://nixos.org/channels/nixos-17.03/nixexprs.tar.xz`
- Using latest channel, but hosted by github: `https://github.com/NixOS/nixpkgs-channels/archive/nixos-17.03.tar.gz`
- Using latest commit for release branch, but not tested yet: `https://github.com/NixOS/nixpkgs/archive/release-17.03.tar.gz`

Examples:

- `nix-build -I ~/dev`
- `nix-build -I ~/dev`
- `nix-build -I nixpkgs=http://nixos.org/channels/nixos-17.03/nixexprs.tar.xz`
- `NIX_PATH=nixpkgs=http://nixos.org/channels/nixos-17.03/nixexprs.tar.xz nix-build ...`
- Using just Nix:

```
with import (fetchTarball https://github.com/NixOS/nixpkgs-channels/archive/nixos-14.
↪12.tar.gz) {};

stdenv.mkDerivation { ... }
```

How to build reverse dependencies of a package?

nox-review wip

I'm getting: writing to file: Connection reset by peer

Too big files in src, out of resources (HDD space, memory)

What are channels and different branches on github?

Subquestion: how stable is unstable?

How do I mirror tarballs?

We have a content-addressed tarball mirror at tarballs.nixos.org for this purpose. "fetchurl" will automatically use this mirror to obtain files by hash. However:

- The mirroring script was not running lately. I've revived it so 16.03 tarballs are mirrored now (https://github.com/NixOS/nixos-org-configurations/commit/a17ccf87deae4fb86639c8d34ab5938edd68d8c4).
- The mirroring script only copies tarballs of packages in the Nixpkgs Hydra jobset. Since moreutils is not part of the jobset, it's not mirrored. This can be fixed by adding a meta.platforms attribute to moreutils.

How can I manage dotfiles in $HOME with Nix?

See following solutions:

- https://github.com/sheenobu/nix-home

Are there some known impurities in builds?

Yes.

- CPU (we try hard to avoid compiling native instructions, but rather hardcode supported ones)
- current time/date
- FileSystem (ext4 has a known bug creating empty files on power loss)
- Kernel
- Timing behaviour of the build system (parallel Make not getting correct inputs in some cases)

What's the recommended process for building custom packages?

E.g. if I git clone nixpkgs how do I use the cloned repo to define new / updated packages?

NixOS

How to build my own ISO?

http://nixos.org/nixos/manual/index.html#sec-building-cd

How do I connect to any of the machines in NixOS tests?

Apply following patch:

```
diff --git a/nixos/lib/test-driver/test-driver.pl b/nixos/lib/test-driver/test-driver.
↪pl
index 8ad0d67..838fbdd 100644
--- a/nixos/lib/test-driver/test-driver.pl
+++ b/nixos/lib/test-driver/test-driver.pl
@@ -34,7 +34,7 @@ foreach my $vlan (split / /, $ENV{VLANS} || "") {
     if ($pid == 0) {
         dup2(fileno($pty->slave), 0);
         dup2(fileno($stdoutW), 1);
-        exec "vde_switch -s $socket" or _exit(1);
+        exec "vde_switch -tap tap0 -s $socket" or _exit(1);
     }
     close $stdoutW;
     print $pty "version\n";
```

And then the vde_switch network should be accessible locally.

How to bootstrap NixOS inside an existing Linux installation?

There are a couple of tools:

- https://github.com/jeaye/nixos-in-place
- https://github.com/elitak/nixos-infect
- https://github.com/cleverca22/nix-tests/tree/master/kexec

Hydra

What to do if cache/hydra is down or unreachable?

Pass following to Nix commands:

- `--option connect-timeout 5` to wait only 5 second on binary package
- `--fallback` to build from source if binary package fetching fails

How do I add a new binary cache?

Using *NixOS*:

```
trustedBinaryCaches = [ "https://cache.nixos.org" "https://hydra.snabb.co" ];
binaryCaches = trustedBinaryCaches;
binaryCachePublicKeys = [ "hydra.snabb.co-1:zPzKSJ1mynGtYEVbUR0QVZf9TLcaygz/
↪OyzHlWo5AMM=" ];
```

Using *Nix*:

```
$ echo "trusted-binary-caches = https://hydra.snabb.co" >> /etc/nix/nix.conf
$ nix-build helpers/bench.nix --option extra-binary-caches https://hydra.snabb.co`
```

Software Nix has influenced

- https://github.com/alexanderGugel/ied
- https://www.habitat.sh/
- https://www.gnu.org/software/guix/

Contributing to the cookbook

Guidelines

http://creativecommons.org/licenses/by/4.0/

Getting the source code

Building documentation

Submitting a pull request