# Contents

1	nix- (nix original command)	1
2	nix (nix command improved)	2
3	nixos	2
4	flake           4.1 flake override	<b>2</b> 3
5	nix-shell	3
6	nixpkgs6.1 confirm packages for nix-build, nix-instantiate, nix-channel6.2 confirm packages for nix-env6.3 confirm packages for flakes	3 3 4 4
7	nix repl 7.1 nix-repl> 명령어	<b>4</b>
8	pictures	6
1	nix- (nix original command)	
	\$nix-env –list-generations \$nix-env -G 42 or \$nix-env –switch-generation 42 \$nix-env –rollback	
	\$nix-env -e firefox \$nix-env -e '.*' \$nix-store -delete \$nix-store -gc \$nix-collect-garbage -d	

# 2 nix (nix command improved)

\$nix-env 대응 nix profile

\$nix-env -iA firefox \$nix profile install firefox \$nix profile install nixpkgs#hello

\$nix-env -e firefox \$nix profile remove firefox

\$nix-env -q \$nix profile list \$nix-env -list-generations \$nix profile history \$nix-env -rollback \$nix profile rollback

\$nix-env -qa \$nix-env -qaPs

\$nix profile install \$nix profile remove \$nix profile rollback

\$nix profile list 
\$nix profile history

nix-env로 설치한 App의 profile은 manifest.nix 파일로 관리되지만, nix profile 로 설치한 App은 manifest.json으로 관리된다. nix profile로 App을 설치하면 profile의 관리 파일 자체가 바뀌기 때문에 더 이상 nix-env로 설치할 수 없다.

\$nix-store -q -tree 'realpath ./result/bin/S4 \$nix-store -q -referrers-closure 'realpath ./result/bin/S4

## 3 nixos

\$nixos-rebuild switch -rollbackflake.nix가 있는 폴더>flake.nix가 있는 폴더>sudo nixos-rebuild switch -flake .#syryuhds -impure

#### 4 flake

[flake.nix가 있는 폴더]\$nix build nix build github:syryuauros/myflakes/masterflake.nix가 있는 폴더>flake.nix가 있는 폴더>flake.nix가 있는 폴더>flake.nix가 있는 폴더

\$ nix run nixpkgs#gnuplot

\$realpath ./result/bin/S4 \$nix-store -q -tree 'realpath ./result/bin/S4'

\$nix flake metadata github:edolstra/dwarffs

\$nix flake show github:edolstra/dwarffs

\$nix-store -q -referrers-closure 'realpath ./result/bin/S4

\$ nix registry add nixpkgs github:NixOS/nixpkgs/5272327b81ed355bbed5659b8d303cf2979b6953

\$nix-info -m

"https://github.com/<owner>/<project>/commit/<hash>" example

https://github.com/nixos/nixpkgs/commit/ede02b4ccb13557b95058d66146640a2b0bb198f

\$nix build (default package만 설치) \$nix build .#s4 .#gnuplot (default외에 다른 package설치를 위해서는 직접 package명을 입력해야 한다

#### 4.1 flake override

```
ex) (openblas.override { singleThreaded = true;})
or let openblas-single-thread = openblas.override { singleThreaded = true; };
```

#### 5 nix-shell

nix-shell -p 'lua.withPackages(ps: with ps; [ busted luafilesystem ])'

#### 6 nixpkgs

### 6.1 confirm packages for nix-build, nix-instantiate, nix-channel

- 1 \$nixrepl [Ret] nix-repl>:l <nixpkgs> [Ret] nix-repl>keyword(ex:rust) + TAB
- $2 \quad /\text{nix/var/nix/profiles/per-user/root/channels/nixos/pkgs/top-level/all-packages.nix}$
- 3 \$cd 'nix-instantiate -eval -E '<nixpkgs>' [Ret] cat > all-packages.nix
- 4 \$NIX\_PATH, 경로 확인 후 이동해서 all-packages.nix 확인

### 6.2 confirm packages for nix-env

- 1 \$nix-env -qa 'keyword.\*' or \$nix-env -qaPs 'keyword'
- 2 \$cd ~/.nix-defexpr/channels\_root/nixos/pkgs/top-level, all-packages.nix

#### 6.3 confirm packages for flakes

1 flake.lock 파일 확인 » "nixpkgs" 항목의 "rev" term 확인 » github.com/[owner]/[repo]/tree/[rev]

그림 참조

# 7 nix repl

### 7.1 nix-repl> 명령어

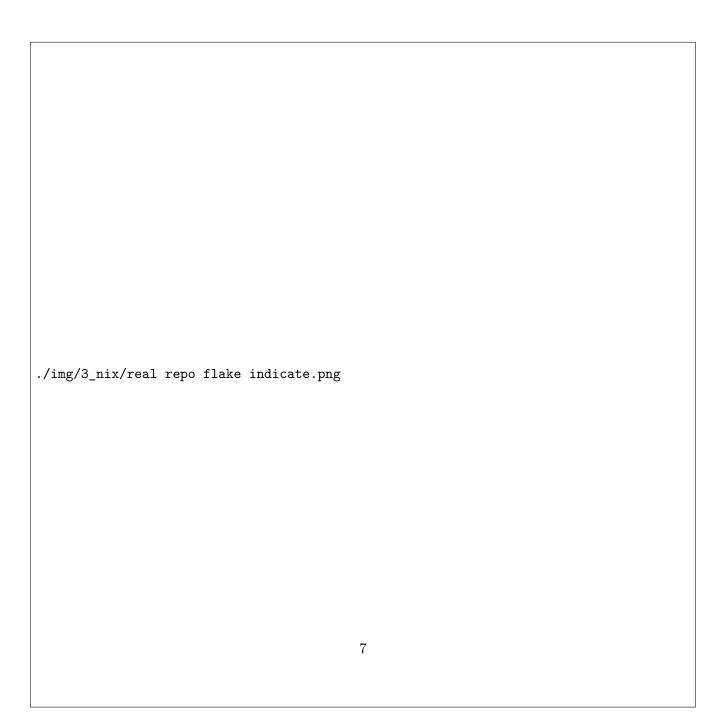
 :help
 도움말

 입력 중간에 TAB
 자동 완성

:doc keyword 해당 명령어 도움말

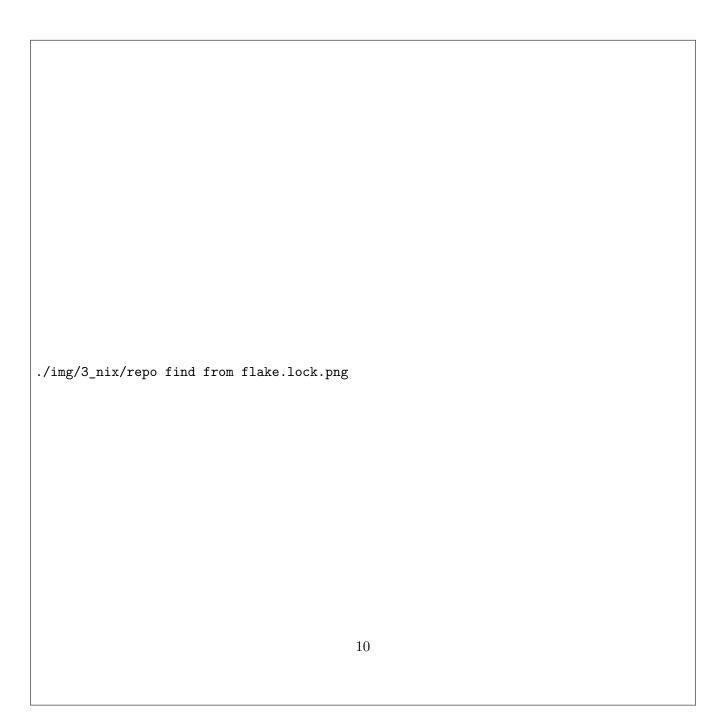
# 8 pictures

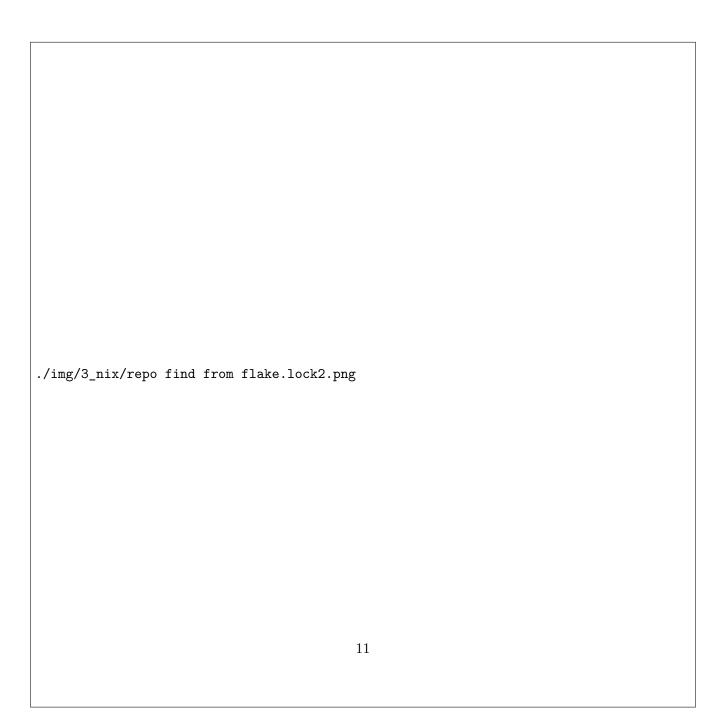
./img/3\_nix/flake input url.png

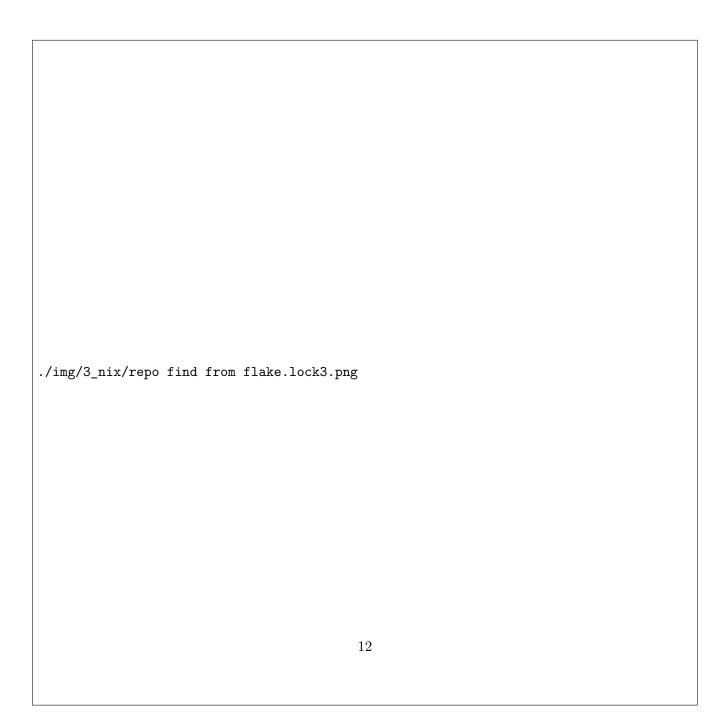


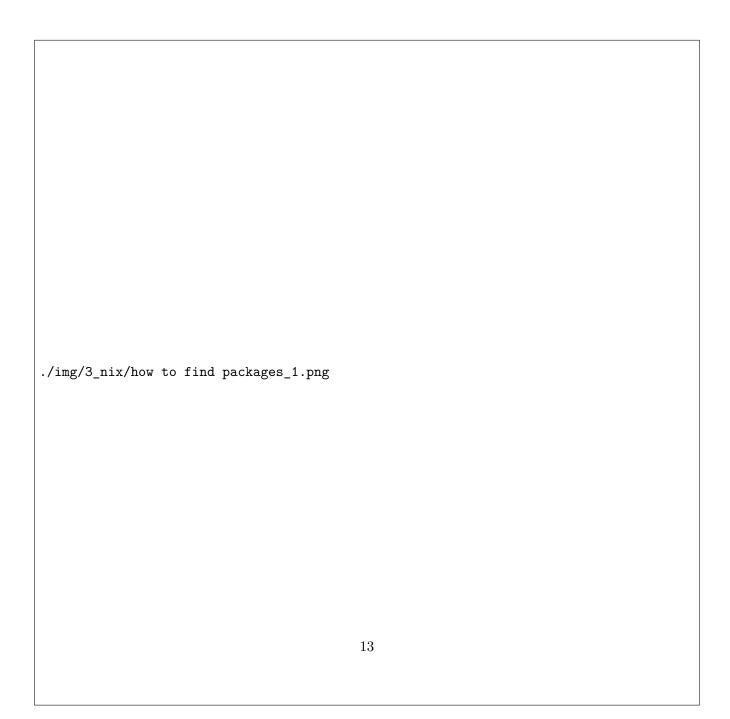


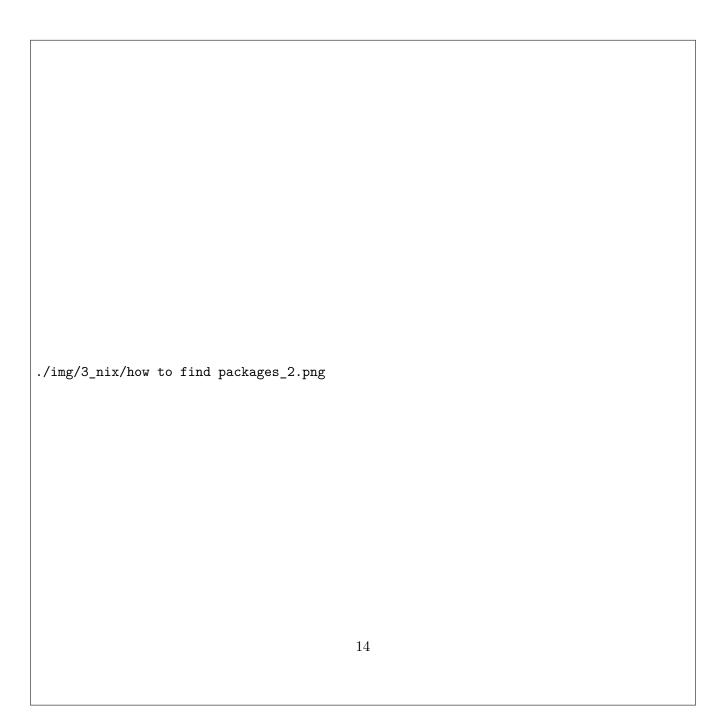












.