NixOS Cheatsheet

Revision 2025-12-27, made by SIPB

https://forgejo.mit.edu/SIPB/nixos-cheatsheet

Task	Command
Running software	
Search for software	`nix search nixpkgs hello`[1]
Run program	`nix run nixpkgs#hello`[2]
Run program from other nixpkgs branch	`nix run nixpkgs/nixos-25.11#hello`
Run program from another flake	`nix run github:user/nixpkgs/branch#hello`
Run specific version of a program	See nixhub.io
Get shell with program	`nix shell nixpkgs#hello`
Run AppImage	`appimage-run hello.AppImage`
Building the system	
Update flake inputs	`nix flake update`
Build and deploy to current machine	`nixos-rebuild switch --sudo`
Build remotely, deploy to current machine	`sudo nixos-rebuild switch --build-host user@remote --use-substitutes`[3]
Build on current machine, deploy remotely	`nixos-rebuild switch --flake .#remote-machine-hostname --target-host root@remote`[4]
Build and run VM from config	`nixos-rebuild build-vm && result/bin/run-$hostname-vm -m 4G -smp 4 -device virtio-vga-gl -display sdl,gl=on`[5]
Build ISO from config	Add `"/installer/cd-dvd/installation-cd-minimal-new-kernel-no-zfs.nix"` as a module, then run `nix build .#nixosConfigurations.$hostname.config.system.build.isoImage`
System generations	
List system generations	`nixos-rebuild list-generations`
Get changes between system generations	`nix profile diff-closures --profile /nix/var/nix/profiles/system`
Roll back one generation	`nixos-rebuild switch --sudo --rollback`
Switch to generation	`sudo nix-env --switch-generation 12345 -p /nix/var/nix/profiles/system && sudo /nix/var/nix/profiles/system/bin/switch-to-configuration switch`
Store operations	
Get size of system	`nix path-info -Sh /nix/var/nix/profiles/system`
Get size of Nix store	`sqlite3 /nix/var/nix/db/db.sqlite 'SELECT SUM(narSize) FROM ValidPaths'`
Get largest packages	`sqlite3 /nix/var/nix/db/db.sqlite 'SELECT * FROM ValidPaths ORDER BY narSize DESC LIMIT 20'`
Clean Nix store	`sudo nix-collect-garbage && nix store optimise`
Delete generations older than 7 days	`sudo nix-collect-garbage --delete-older-than 7d`
Check what depends on path	`nix-store --query --referrers /nix/store/aaaaaa`
Nix development	
Run Nix file	`nix eval -f main.nix`
Inspect system config in REPL	`nix repl .#nixosConfigurations.$hostname`
Format flake	Add `formatter.${system} = pkgs.nixfmt-tree;` to your flake outputs, then run `nix fmt`

[1] This evaluates nixpkgs so the first run may be slow, but further runs used cached results.

[2] Pass `--` before any arguments to the program.

[3] This needs `sudo` to add untrusted paths to the Nix store, since `--sudo` only uses root for activation.

[4] This needs root on the remote machine to add untrusted paths to the remote machine's Nix store.

[5] You may need to change all occurrences of the QEMU version in the run script if the config uses a different nixpkgs revision than the host.