



NixOS Cheatsheet

Made by SIPB

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

Task	Command
Running software	
Search for software	<code>nix search nixpkgs hello</code>
Run program	<code>nix run nixpkgs#hello¹</code>
Run program from other nixpkgs branch	<code>nix run nixpkgs/nixos-25.11#hello</code>
Run program from another flake	<code>nix run nix run github:user/nixpkgs/branch#hello</code>
Find nixpkgs revision for a specific version of a program	See https://lazamar.co.uk/nix-versions/
Get shell with program	<code>nix shell nixpkgs#hello</code>
Run AppImage	<code>appimage-run hello.AppImage</code>
Building the system	
Update flake inputs	<code>nix flake update</code>
Build and deploy to this machine	<code>nixos-rebuild switch --sudo</code>
Build on remote machine and deploy to current machine	<code>sudo nixos-rebuild switch --build-host user@remote --use-substitutes²</code>
Build on current machine and deploy to remote machine	<code>nixos-rebuild switch --flake .#remote-machine-hostname --target-host root@remote³</code>
Build and run VM from config	<code>nixos-rebuild build-vm && result/bin/run-\$hostname-vm -m 4G -smp 4 -device virtio-vga-gl -display sdl,gl=on⁴</code>

¹Pass -- before any arguments to the program

²This needs root on the current machine to add untrusted paths to the Nix store

³This needs root on the remote machine for the same reason

Task	Command
Build ISO from config	Add <code>"/installer/cd-dvd/installation-cd-minimal-new-kernel-no-zfs.nix"</code> as a module, then run <code>nix build .#nixosConfigurations.\$hostname.config.system.build.isoImage</code>
System generations	
List system generations	<code>nixos-rebuild list-generations</code>
Get changes between system generations	<code>nix profile diff-closures --profile /nix/var/nix/profiles/system</code>
Roll back one generation	<code>nixos-rebuild switch --sudo --rollback</code>
Switch to generation	<code>sudo nix-env --switch-generation 12345 -p /nix/var/nix/profiles/system && sudo /nix/var/nix/profiles/system/bin/switch-to-configuration switch</code>
Store operations	
Get size of system	<code>nix path-info -Sh /nix/var/nix/profiles/system</code>
Get size of Nix store	<code>sqlite3 /nix/var/nix/db/db.sqlite 'SELECT SUM(narSize) FROM ValidPaths'</code>
Get largest packages	<code>sqlite3 /nix/var/nix/db/db.sqlite 'SELECT * FROM ValidPaths ORDER BY narSize DESC LIMIT 20'</code>
Clean Nix store	<code>sudo nix-collect-garbage && nix store optimise</code>
Delete generations older than 7 days	<code>sudo nix-collect-garbage --delete-older-than 7d</code>
Check what depends on path	<code>nix-store --query --referrers /nix/store/aaaaaa</code>
Writing Nix code	
Run Nix file	<code>nix eval -f main.nix</code>
Inspect system config in REPL	<code>nix repl .#nixosConfigurations.\$hostname</code>
Format flake	Add <code>formatter.\${system} = pkgs.nixfmt-tree;</code> to your flake outputs, then run <code>nix fmt</code>

⁴You 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