

# Automatic environment activation with direnv

Instead of manually activating the environment for each project, you can reload a [declarative shell](#) every time you enter the project's directory or change the `shell.nix` inside it.

1. Make nix-direnv available
2. Hook it into your shell

For example, write a `shell.nix` with the following contents:

```
1 let
2   nixpkgs = fetchTarball "https://github.com/NixOS/nixpkgs/tarball/nixos-23.11";
3   pkgs = import nixpkgs { config = {}; overlays = []; };
4 in
5
6 pkgs.mkShellNoCC {
7   packages = with pkgs; [
8     hello
9   ];
10 }
```

From the top-level directory of your project run:

```
$ echo "use nix" > .envrc && direnv allow
```

The next time you launch your terminal and enter the top-level directory of your project, `direnv` will automatically launch the shell defined in `shell.nix`

```
$ cd myproject
$ which hello
/nix/store/1gxz5nfzfnhyxjdyzi04r86sh61y4i00-hello-2.12.1/bin/hello
```

`direnv` will also check for changes to the `shell.nix` file.

Make the following addition:

```
1 +
```

[Skip to main content](#)

```
pkgs = import nixpkgs { config = {}; overlays = []; };  
in  
  
pkgs.mkShellNoCC {  
  packages = with pkgs; [  
    hello  
  ];  
+  
+  shellHook = ''  
+    hello  
+    '';  
}
```

The running environment should reload itself after the first interaction (run any command or press `Enter`)

```
Hello, world!
```