

# Getting Started with Nix

**A very brief introduction to Nix(OS)**

# The Basics

## What is Nix?

- **Functional Programming Language**
  - Not general-purpose, “only” for defining packages

# The Basics

## What is Nix?

- **Functional Programming Language**
  - Not general-purpose, “only” for defining packages
- **Package Manager**
  - Packages defined by Nix expressions

# The Basics

## What is Nix?

- **Functional Programming Language**
  - Not general-purpose, “only” for defining packages
- **Package Manager**
  - Packages defined by Nix expressions
- **Linux Distribution (NixOS)**
  - Uses the Nix package manager
  - Configure your system via Nix expressions

# The Basics

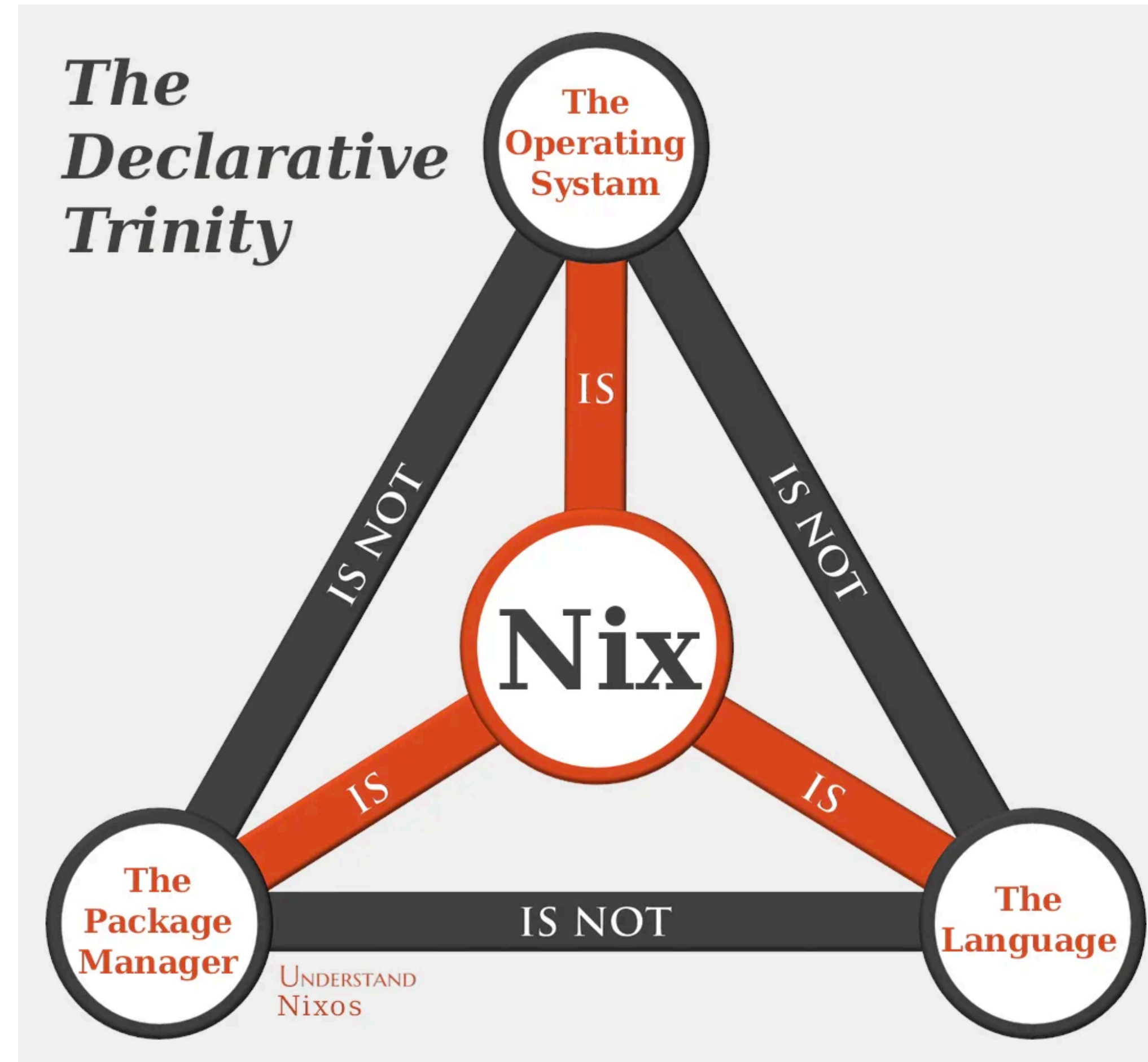
## What is Nix?

- **Functional Programming Language**
  - Not general-purpose, “only” for defining packages
- **Package Manager**
  - Packages defined by Nix expressions
- **Linux Distribution (NixOS)**
  - Uses the Nix package manager
  - Configure your system via Nix expressions

**All are referred to as  
“Nix”**

# The Basics

## What is Nix?



# The Powers of Nix

## Why even bother?

- Largest package repo (80k+ packages)
- Nix builds are
  - Declarative
  - Reproducible
  - Pure
- Manage your entire system declaratively (with NixOS)

# The Nix Language

## How do you write Nix?

- **Not** a general-purpose language
- Purely functional
- Lazy
- Dynamically typed

```
nix-repl> x = 4

nix-repl> double = a: a*2

nix-repl> double 4
8

nix-repl> add = a: b: a + b

nix-repl> add 2 3
5

nix-repl> mul = {a, b}: a * b

nix-repl> mul { a = 2; b = 3; }
6
```



# The Nix Package Manager

## How do you build and run things with Nix?

- Install system-wide packages
- Spawn a temporary shell with the package
- Flakes

```
~ > nix profile install nixpkgs#hello  
  
~ > hello  
Hello, world!
```

```
~ > nix shell nixpkgs#hello  
  
~ > hello  
Hello, world!
```

# NixOS

## How do you Nix-ify all the things?

- Have one system configuration (Collection of Nix files)
- Manage your whole system with Nix
  - Installed packages
  - Program configuration (*Dotfiles*)
- Setup new machines with ease
- Rollback to old generations

```
1 { ... }:{
2     imports = [
3         ./plugins
4     ];
5     programs.neovim = {
6         enable = true;
7         vimAlias = true;
8         defaultEditor = true;
9         vimdiffAlias = true;
10    };
11 }
```

# Appendix: Flakes

## An easy way to share Nix code

- The “new way” of using Nix
- Bundle of Flake with code and Lockfile for inputs
- Flakes can import other Flakes
- Nixpkgs is a Flake
- **Reality: Everyone uses Flakes for Everything**

# Appendix: Flakes

# An easy way to share Nix code

# Flake Reference

# Input

```
~ > nix run github:snowfallorg/cowsay -- test
```

-----  
< test >  
-----  
 \   
 / \ ~~~ / \   
 ( .. ) ----- ,   
 \\_\_ \_\_/   
 ) | / ) \ \   
 | /\ /\_\_ \ / ^   
 "-|\_|\_|\_|\_|

**Now go and install Nix(OS)! ❄️**

(If you want to)