

## Why choose Nix or NixOS?

### It's Reproducible...

Nix builds packages in isolation from each other. This ensures that they are reproducible and don't have undeclared dependencies, so if a package works on one machine, it will also work on another.

### It's Declarative...

Nix makes it trivial to share development and build environments for your projects, regardless of what programming languages and tools you're using.

### It's Reliable...

Nix ensures that installing or upgrading one package cannot break other packages. It allows you to roll back to previous versions, and ensures that no package is in an inconsistent state during an upgrade.

## Why choose Nix or NixOS?

### It's Reproducible...

Nix builds packages in isolation from each other. This ensures that they are reproducible and don't have undeclared dependencies, so if a package works on one machine, it will also work on another.

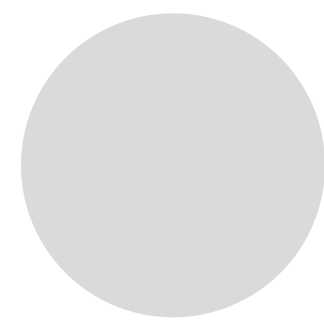
### It's Declarative...

Nix makes it trivial to share development and build environments for your projects, regardless of what programming languages and tools you're using.

### It's Reliable...

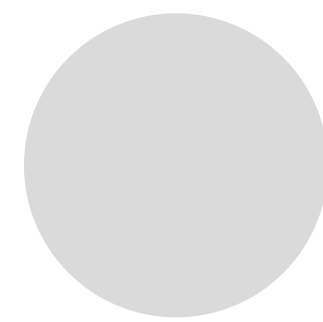
Nix ensures that installing or upgrading one package cannot break other packages. It allows you to roll back to previous versions, and ensures that no package is in an inconsistent state during an upgrade.

## Why choose Nix or NixOS?



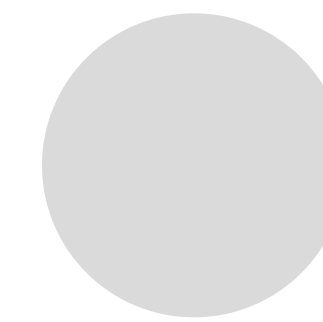
### It's Reproducible...

Nix builds packages in isolation from each other. This ensures that they are reproducible and don't have undeclared dependencies, so if a package works on one machine, it will also work on another.



### It's Declarative...

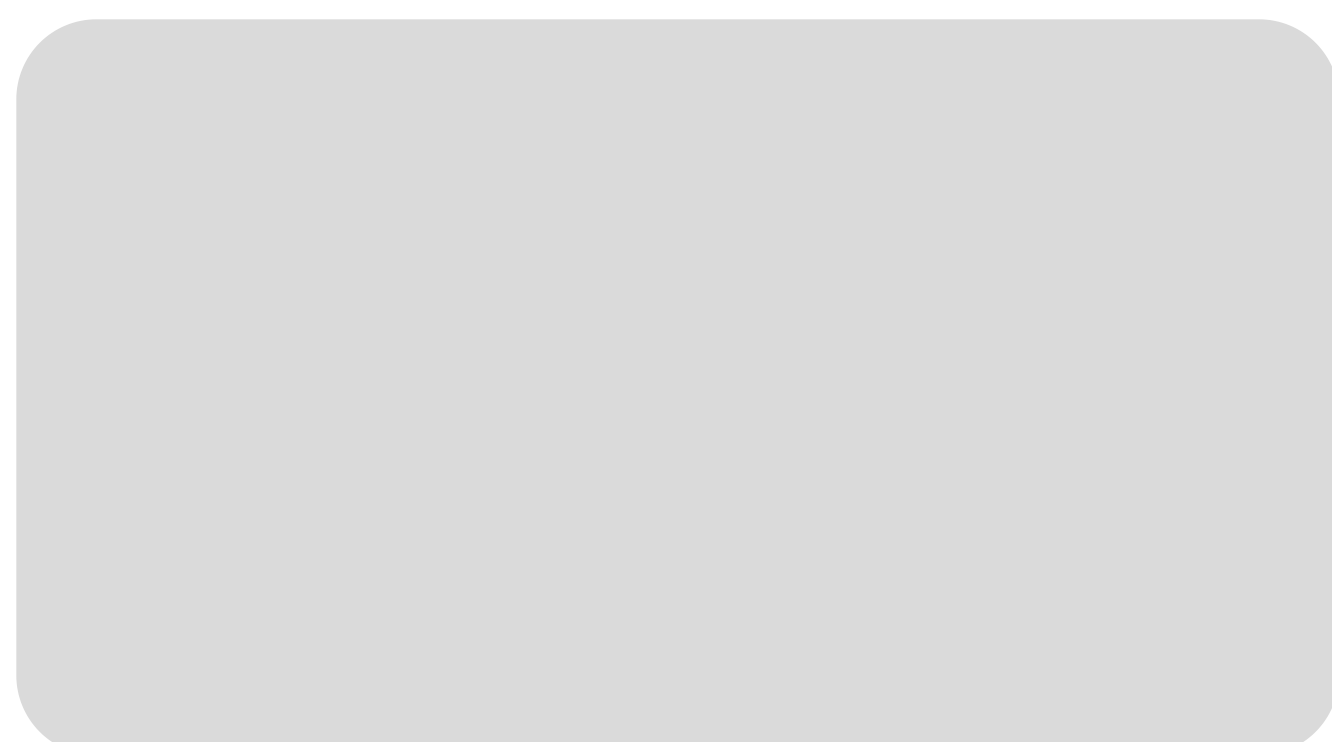
Nix makes it trivial to share development and build environments for your projects, regardless of what programming languages and tools you're using.



### It's Reliable...

Nix ensures that installing or upgrading one package cannot break other packages. It allows you to roll back to previous versions, and ensures that no package is in an inconsistent state during an upgrade.

## Why choose Nix or NixOS?



### It's Reproducible...

Nix builds packages in isolation from each other. This ensures that they are reproducible and don't have undeclared dependencies, so if a package works on one machine, it will also work on another.

More

