NIXTCLOUD

Self-Hosted Cloud in One Command

Patras Tech Talk | Jan 2026 | PWC

Vaios Karastathis

- ⭐ github.com/jjacke13/nixtcloud (18 stars)
- 💼 linkedin.com/in/vaios-karastathis
- 📺 YouTube: @eikosiena

Where Is Your Data?

The Problem

☁️ Cloud Services

- Your data on their servers
- No control, no privacy
- Terms you never read
- Data analyzed & monetized

🏠 Self-Hosting

- Complex setup (ports, DNS, SSL)
- Constant maintenance
- Manual updates & security
- Fragile, error-prone

We need a better way.

What If...

```
$ nix build github:jjacke13/nixtcloud
```

One command. Flash to SD-card/Disk. Boot. Done.

Welcome to **NIXTCLOUD**

The Three Pillars

NixOS

Declarative Foundation

Nextcloud

Cloud Application

Holesail

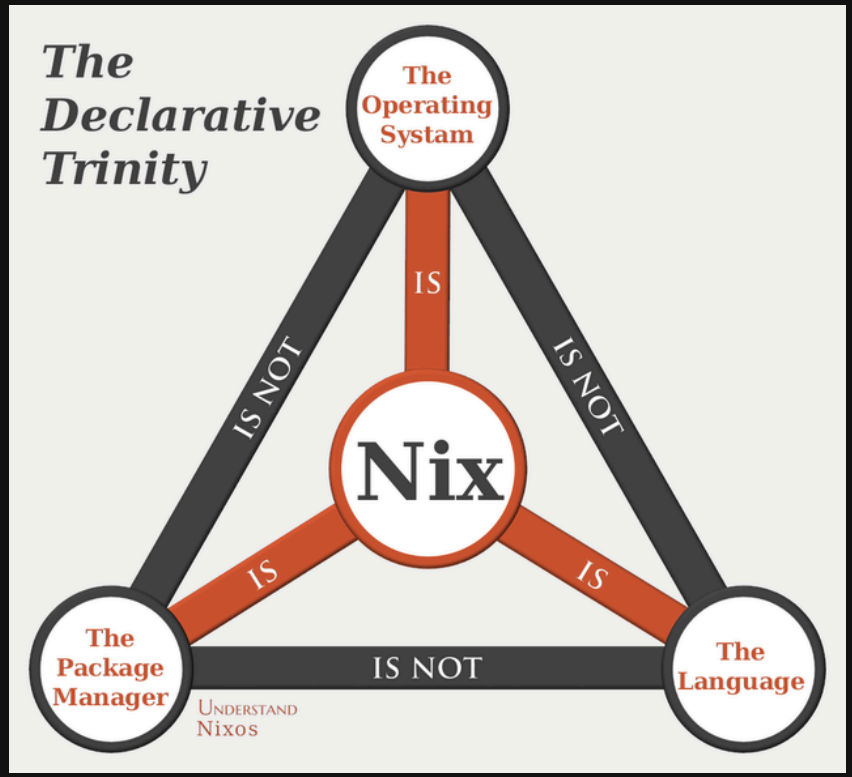
P2P Connectivity

Each pillar solves a critical problem. Together, they create something greater.

Pillar 1

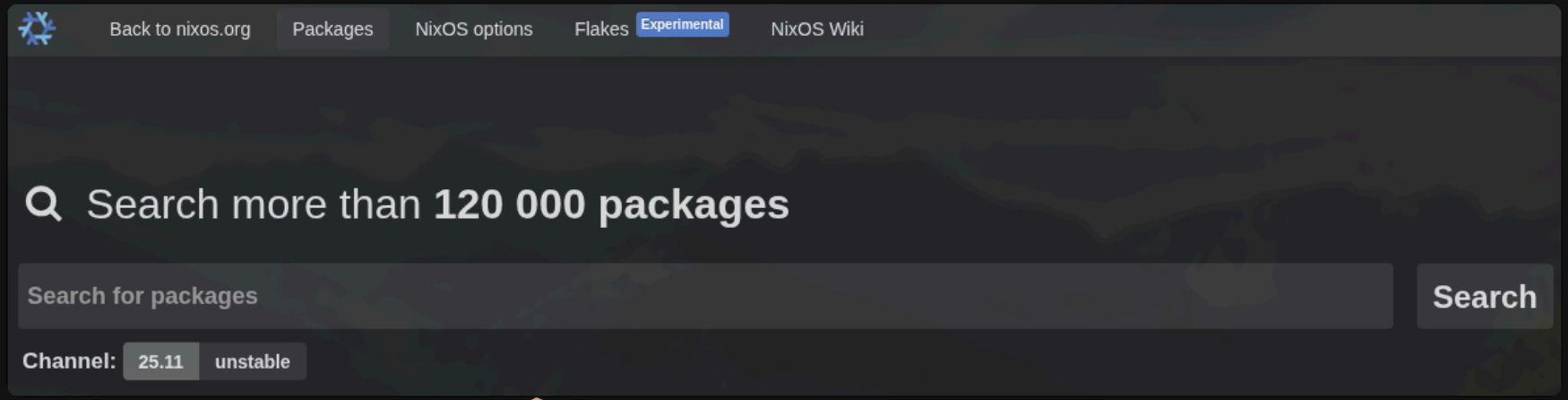
NixOS — The Foundation

What is Nix?



- Functional Language
  - Pure functional approach to package management
- Package Manager
  - Declarative package definitions and builds
  - (Linux - MacOS - WSL)
  - (x86 / aarch64 / i686)
- NixOS
  - Complete Linux distribution built on Nix principles

The Nixpkgs Ecosystem



📦 120,000+ packages

The largest package repository

github.com/NixOS/nixpkgs

Nix Power Tools

🛠️ nix-shell

Instant development environments

```
# Need Python + numpy?
nix-shell -p python3 python3Packages.numpy

# Or declaratively:
nix-shell shell.nix
```

- Zero system pollution
- Reproducible across machines
- Perfect for DevOps workflows

🏗️ nix-build

Declarative builds

```
  pkgs.stdenv.mkDerivation {
    pname = "hello-cpp";
    version = "1.0.0";
    src = ./.;
    buildInputs = [ pkgs.cowsay ];
    nativeBuildInputs = [ pkgs.gcc ];
    buildPhase = ''
       g++ -o hello simple.cpp
    '';

    installPhase = ''
    mkdir -p $out/bin
    cp hello $out/bin/
  '';
}
```

- Strictly defined dependencies

Why NixOS?

Traditional Linux

- **Imperative** — "Run these commands"
- **Mutable** — System changes in place
- **Configuration Drift** — "It worked last month..."
- **Manual Rollback** — Hope you have backups

NixOS

- **Declarative** — "This is the desired state"
- **Immutable** — System rebuilt from config
- **Reproducible** — Same config = Same system
- **Atomic Generations** — Boot previous instantly

NixOS is Infrastructure as Code taken to its logical conclusion.

NixOS Configuration Example

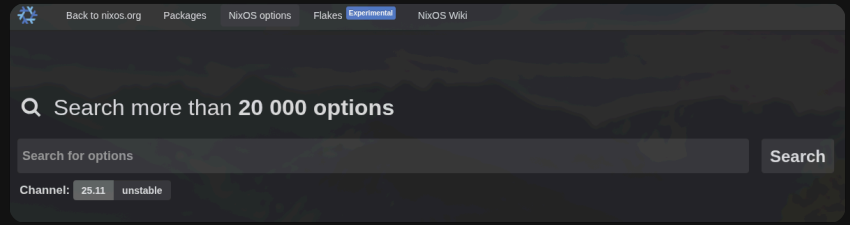
```
{
  boot.loader.grub.device = "/dev/sda";

  networking.hostName = "myserver";
  networking.firewall.enable = true;

  services.nginx.enable = true;

  users.users.admin = {
    isNormalUser = true;
    extraGroups = [ "wheel" ];
  };

  environment.systemPackages = [
    pkgs.htop
    pkgs.git
  ];
}
```



Your entire system defined in one file. Change it, rebuild, done.

Reproducibility: The Superpower

configuration.nix

Your config

IDENTICAL IMAGE — Every. Single. Time.

- ✅ Build on your laptop → runs on Raspberry Pi
- ✅ Build today → build next year → same result
- ✅ Share config → anyone can reproduce your system

Atomic Upgrades & Instant Rollback

Boot Menu:

```
NixOS - Generation 47 (current)     ← Latest

NixOS - Generation 46               ← Previous

NixOS - Generation 45

NixOS - Generation 44

...
```

Update broke something?

→ Reboot, select previous generation
→ System restored in **30 seconds**

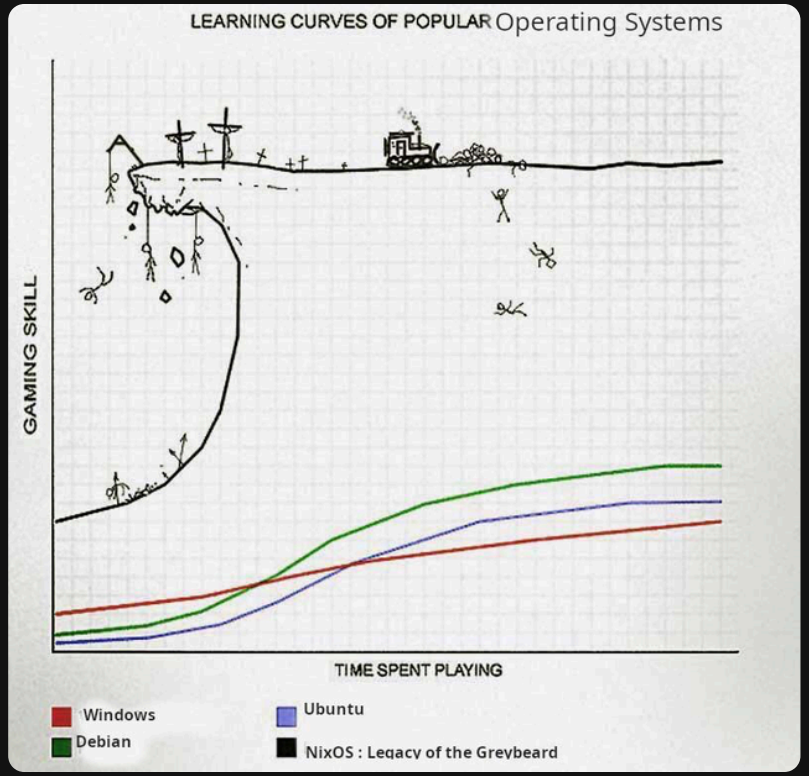
No reinstallation. No lost weekends.

Why This Matters for Self-Hosting

- ✅ No Configuration Drift
  - Your system stays exactly as defined
- ✅ Reproducible Builds
  - Rebuild your server from scratch anytime
- ✅ Easy Recovery
  - Rollback to working state instantly
- ✅ Shareable Infrastructure
  - Your config IS the documentation
- ✅ Auditable
  - Review every package, every service

Your machine becomes a server you can TRUST.

What’s the Catch?



Pillar 2

Nextcloud — The Application

Nextcloud: Your Private Cloud

☁️ Open-source cloud platform

400,000+ deployments worldwide

Feature	Description
📁 Files	Sync & share
📅 Calendar	Manage schedule
👥 Contacts	Address book
📝 Notes	Markdown notes
🔌 200+ Apps	Extend it

Pre-Configured & Ready

Zero Configuration Required:

1. Connect ethernet & power
2. Boot your device
3. Wait 5 minutes for first-boot setup
4. Visit `https://nixtcloud.local`
5. Login: `admin` / `admin`
6. Change your password

- ✅ HTTPS enabled by default
- ✅ Database pre-configured
- ✅ Cron jobs scheduled
- ✅ External storage ready (NTFS supported ☺)

```
services.nextcloud = {
  enable = true;
  package = pkgs.nextcloud32;
  hostName = name;
  database.createLocally = true;
  config = {
    dbtype = "sqlite";
    adminuser = "admin";
    adminpassFile = "/etc/nixos/adminpass.txt";
  };
  settings = {
    trusted_domains = [
      "localhost"
      "${name}.local"
      "192.168.*.*"
    ];
    default_phone_region = "GR";
    log_type = "file";
    loglevel = 4;
    maintenance_window_start = 1;
    quota_include_external_storage = true;
    overwriteprotocol = "https";
    overwritecondaddr = "^192\\.168\\..*$";
```

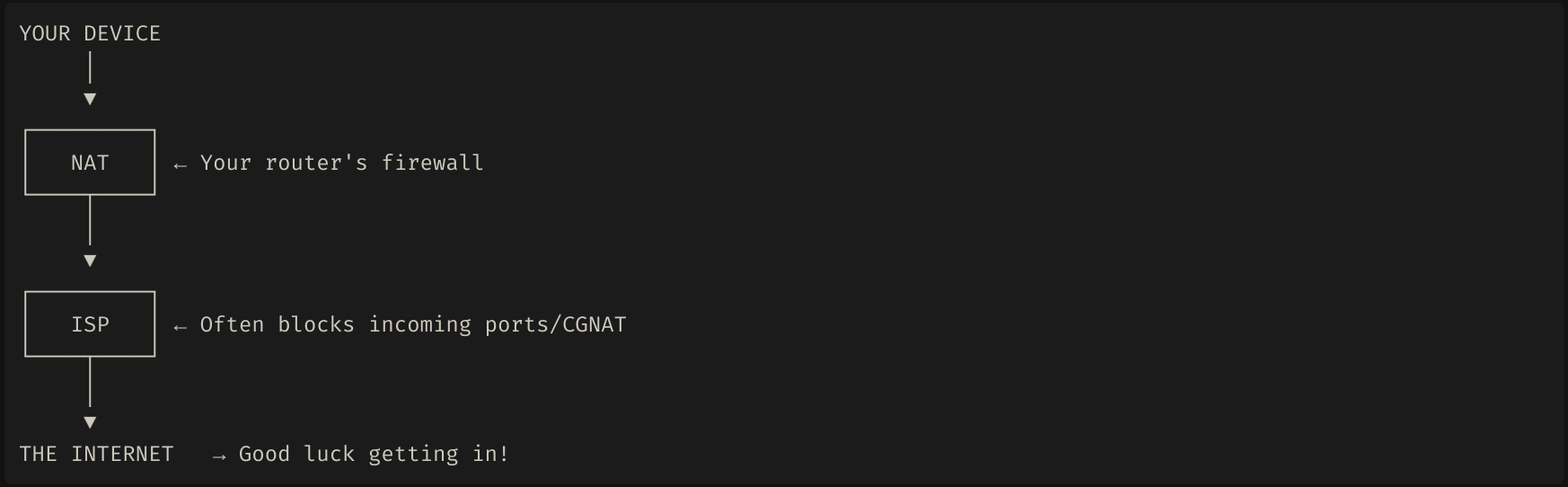
Remote acces???

Pillar 3

Holesail — The Magic

The Connectivity Problem

How do you access your home server from outside?



- **Port forwarding?** Router settings vary wildly.
- **Dynamic DNS?** Your IP changes constantly.
- **VPN services?** Another subscription, another company.

Enter Holesail

"Peer-to-peer tunnels for everyone"

- ✅ No port forwarding
- ✅ No dynamic DNS
- ✅ No central server
- ✅ No account required
- ✅ End-to-end encrypted

Just a CONNECTION STRING:

```
$ npm install -g holesail

$ holesail --live 80
```

```
Key: hs://s0003a3d2c1774c336e2c18828662b7b7c66
```

Share the string → Connect from anywhere

Built on Pear Runtime

Pear Runtime - P2P application platform by Holepunch.to

Keet.io by Holepunch

Keet - P2P Chat

- Encrypted chat & video
- No servers, no tracking
- Built on Pear, like Holesail

Holesail-nix: My Contribution

github.com/jjacke13/holesail-nix ⭐ 4

MIT License

NixOS Modules:

Module	Purpose
`holesail-server`	Expose local ports
`holesail-client`	Connect to remote
`holesail-filemanager`	Web file share

```
services.holesail-server.mycloud = {
  enable = true;
  port = 80;
  key-file = "/var/lib/key.txt";
};
```

Declarative P2P. The Nix way.

QR-Based Access

In your Nextcloud, Nixtcloud creates:

📄 remote.txt

```
hs://s000abc123def456ghi789jkl012mno ...
```

🖼️ remote.jpg

How to use:

1. **At home:** Scan the QR / save the hs:// link
2. **Traveling:** Open Holesail app
3. **Scan/paste the link**
4. **Connect:** Access your cloud instantly

Privacy by Design

🔐 END-TO-END ENCRYPTION

Your traffic is encrypted before it leaves your device

🚫 NO CENTRAL SERVER

No company sees your connection. No metadata harvesting.

📝 NO ACCOUNT REQUIRED

No email signup. No identity verification. No tracking.

🔓 OPEN SOURCE

Audit the code yourself. Trust through transparency.

TRUE peer-to-peer. YOUR data. YOUR rules.

Why Nixtcloud is Special

Self-Healing System

🔄 Daily

- Scheduled health checks
- Automatic reboot (configurable)

📅 Weekly

- If Repository updates → System updates applied

⚡ On-Demand

- Delete `reboot.txt` → Reboot
- Delete `remote.txt` → New P2P key

Always:

- ✅ USB drives auto-mount (30 sec delay)
- ✅ Services auto-restart on failure

Set it and forget it. It just works.

Supported Hardware

🥧 Raspberry Pi 4

- 2GB / 4GB / 8GB RAM
- Widely available

~€50-80

🥧 Raspberry Pi 5

- 2GB / 4GB / 8GB RAM
- Faster performance

~€60-90

🍊 NanoPi NEO3

- 2GB RAM
- Gigabit Ethernet
- Ultra-compact
- Custom kernel

~€30-40

NanoPi NEO3: The Challenge

Specs:

- **SoC:** Rockchip RK3328
- **CPU:** Quad-core ARM Cortex-A53
- **RAM:** 2GB DDR4
- **Network:** Gigabit Ethernet

The Problem:

- Stock kernel too large
- Long build times
- Bloated image size

The Solution: (NixOS 😉)

- ✅ Custom minimal kernel
- ✅ Device tree configuration
- ✅ U-Boot optimization

Custom Minimal Kernel

Linux Kernel 6.18.3 — Compiled for NanoPi NEO3

Aspect	Standard	Minimal
Drivers	All	Essential only
Filesystems	All	ext4, FAT, NTFS
Network stacks	All	TCP/IP, basics
Debug features	Included	Stripped
Build time	~45 min	**~15 min**
Image size	~4.5 GB	**~2.7 GB**

🎬 DEMO TIME

Let’s see Nixtcloud in action

💻 Laptop

Local access

📱 Mobile

Remote P2P access

Get Involved

⭐ Star the Repos

Repository	Stars
jjacke13/nixtcloud	⭐ 18
jjacke13/holesail-nix	⭐ 4

🔧 Try It

Download → Flash → Boot → Learn

🤝 Contribute

- Report issues
- Submit pull requests
- Spread the word

📢 Previous Talks

- FOSSCOMM 2024
- FOSSCOMM 2025
- Greek NOG

Thank You

Questions?

github.com/jjacke13/nixtcloud

⭐ Star it | 🍴 Fork it | 🤝 Contribute