1. CI/CD Pipeline: [PacketMix]

Owner	FreshlyBakedCake

Repository	Patisserie (default branch: main)

CI/CD System	Spindle a.k.a. Tangled Actions

Scope

This document describes the CI/CD pipeline for **PacketMix**, including triggers, stages, environments, artifacts, quality gates, and operational procedures.

Contents

1. CI/CD Pipeline: [PacketMix] 1
2. Overview 2
   2.1. Goals 2
   2.2. Non-goals 2
   2.3. High-level flow 2
3. Triggers & Branching 2
   3.1. Trigger events 2
   3.2. Branching strategy 2
4. Environments & Promotion 2
   4.1. Promotion rules 2
5. Secrets and Variables 2
   5.1. Secrets 2
   5.2. Key variables (examples) 2
6. Pipeline Stages 3
   6.1. deadnix 3
   6.2. reuse 3
   6.3. packetmix-npins-duplicate-check 3
   6.4. packetmix-treefmt 4
   6.5. packetmix-build 4
   6.6. packetmix-release 5
   6.7. github 6
7. Quality Gates (Centralized) 6
   7.1. Required checks on PR 6
8. Rollback & Recovery 6
   8.1. Rollback strategy 6

8.2. When to rollback 6
9. Appendix 6
9.1. Reference links 6
9.2. Images 7

2. Overview

2.1. Goals

- Deliver **PacketMix** safely and repeatably.
- Enforce quality gates (tests, approvals).
- Provide auditable deployments and fast rollback.

2.2. Non-goals

- [e.g., Build system internals not covered]

2.3. High-level flow

Commit/PR → **Test** → **Build** → **Deploy**

3. Triggers & Branching

3.1. Trigger events

- `push` to `[main]`
- `pull_request` targeting `[main]`

3.2. Branching strategy

- Primary branch: `[main]`
- Release branch: `[release]`

4. Environments & Promotion

Environment	Type	Promotion rule	Notes
Development	ephemeral	On PR	Runs in Nixery engine
Staging	ephemeral	On merge to main	Runs in Nixery engine
Release	shared	Main passes checks	Sets code for release to machines

4.1. Promotion rules

- Development ⟶ Staging: automatic on successful PR to main pipeline.
- Staging ⟶ Production: automatic on main checks passing.

5. Secrets and Variables

5.1. Secrets

- Storage: Bitwarden vault
- Access model: Provisioned on machine run.

5.2. Key variables (examples)

- `KEY_SSH_RELEASE`: SSH Key used for pushing to git

- `KEY_SSH_REMOTE_BUILD`: SSH Key used for accessing the remote builder

6. Pipeline Stages

6.1. deadnix

Purpose: Check for dead nix expressions.

Expected runtime: ~5 seconds

Runner/agent: linux-x64 via nixery container

Triggers:

- PR
- push to main

Inputs:

- Source code

Steps:

- Check for unused nix bindings

Quality gates:

- No dead nix expressions.

6.2. reuse

Purpose: Ensure all files are properly licensed

Expected runtime: ~5 seconds

Runner/agent: linux-x64 via nixery container

Triggers:

- PR
- push to main

Inputs:

- Source code

Steps:

- Check for REUSE compliance

Quality gates:

- All files are properly licensed.

6.3. packetmix-npins-duplicate-check

Purpose: Catch duplicate npins keys.

Expected runtime: ~5 seconds

Runner/agent: linux-x64 via nixery container

Triggers:

- PR

- push to main

Inputs:

- Source code

Steps:

- Check for duplicate npins keys

Quality gates:

- No duplicate keys.

6.4. packetmix-treefmt

Purpose: Check formatting of changed files.

Expected runtime: ~30 seconds

Runner/agent: linux-x64 via nixery container

Triggers:

- PR
- push to main

Inputs:

- Source code
- KEY_SSH_REMOTE_BUILD secret

Steps:

- Get remote builds ssh key.
- Add base system files.
- Ensure files are formatted.

Quality gates:

- All files are formatted.

6.5. packetmix-build

Purpose: Build the project.

Expected runtime: ~1 hour

Runner/agent: linux-x64 via nixery container

Triggers:

- PR
- push to main

Inputs:

- Source code
- KEY_SSH_REMOTE_BUILD secret

Steps:

- Get remote builds SSH key.
- Add base system files.
- Evaluate all systems
- Build all systems

Outputs:

- Build artifacts

Quality gates:

- All systems evaluate successfully.
- All builds succeed.

6.6. packetmix-release

Purpose: Release to systems.

Expected runtime: ~1 hour

Runner/agent: linux-x64 via nixery container

Triggers:

- push to main

Inputs:

- Source code
- KEY_SSH_REMOTE_BUILD secret
- KEY_SSH_REMOTE_RELEASE secret

Steps:

- Get remote builds SSH key.
- Get release push SSH key.
- Add base system files.
- Evaluate all systems.
- Build all systems.
- Push to release

Outputs:

- Build artifacts

Quality gates:

- All systems evaluate successfully.
- All builds succeed.
- Push to release succeeds.

6.7. github

Purpose: Mirror to github.

Expected runtime: ~1 minute

Runner/agent: linux-x64 via nixery container

Triggers:

- push to release

Inputs:

- Source code
- KEY_SSH_GITHUB secret
- KEY_SSH_GITHUB_PACKETMIX secret
- KEY_SSH_GITHUB_SPRINKLES secret

Steps:

- Write github SSH Key.
- Add base system files.
- Push * to GitHub.
- Write packetmix github SSH Key.
- Push packetmix to GitHub.
- Write sprinkles github SSH Key.
- Push sprinkles to GitHub.

Quality gates:

- All pushes to github succeed

7. Quality Gates (Centralized)

7.1. Required checks on PR

- **Pass condition:** All checks succeed (see Figure 1), Human review approved (if required).
- **On failure:** Block merge; require fix or approved exception.

8. Rollback & Recovery

8.1. Rollback strategy

- Primary: Force push release branch to last known good image.

8.2. When to rollback

- Any system fails to start or crash loops.

9. Appendix

9.1. Reference links

- FreshlyBakedCake/Patisserie Repo

9.2. Images

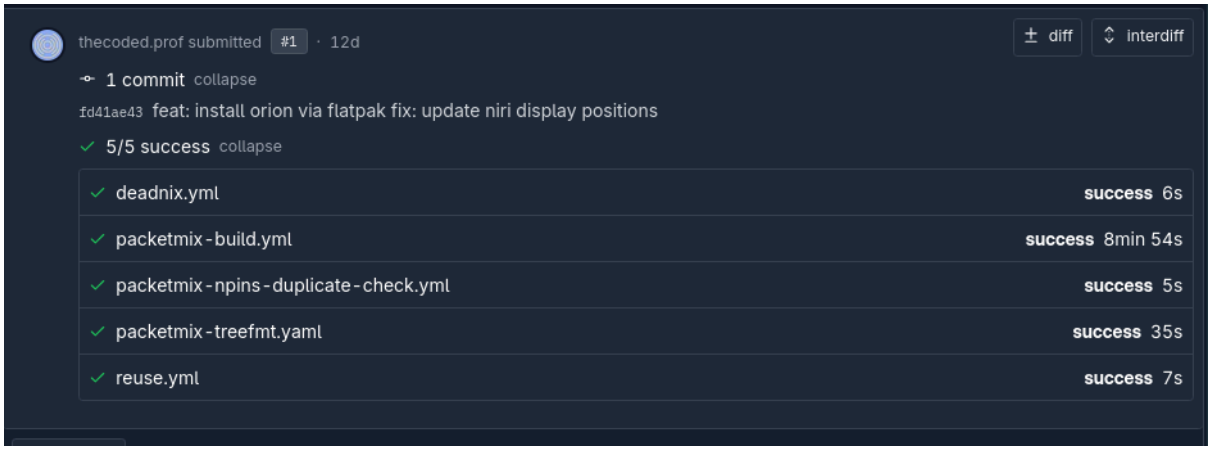


Figure 1: CI Pipeline Success.

```
Jan 23 04:40:18 shorthair systemd[1]: Starting NixOS Upgrade...
   Subject: A start job for unit nixos-upgrade.service has begun execution
   Defined-By: systemd
   Support: https://lists.freedesktop.org/mailman/listinfo/systemd-devel

   A start job for unit nixos-upgrade.service has begun execution.

   The job identifier is 3839.
Jan 23 04:40:21 shorthair nixos-upgrade-pre-start[54628]: HEAD is now at 91394f8 feat(pm/coded): set firefox as default browser
Jan 23 04:40:22 shorthair nixos-upgrade-start[54632]: unpacking 1 channels...
Jan 23 04:40:33 shorthair nixos-upgrade-start[54631]: building the system configuration...
Jan 23 04:40:53 shorthair nixos-upgrade-start[54631]: Done. The new configuration is /nix/store/i04rr3hid8hf0dhc34l6n715c2x457dz-nixos-system-shorthair-25.11.1948.c6f52ebd45e5
Jan 23 04:40:53 shorthair systemd[1]: nixos-upgrade.service: Deactivated successfully.
   Subject: Unit succeeded
   Defined-By: systemd
   Support: https://lists.freedesktop.org/mailman/listinfo/systemd-devel

   The unit nixos-upgrade.service has successfully entered the 'dead' state.
Jan 23 04:40:53 shorthair systemd[1]: Finished NixOS Upgrade.
   Subject: A start job for unit nixos-upgrade.service has finished successfully
   Defined-By: systemd
   Support: https://lists.freedesktop.org/mailman/listinfo/systemd-devel

   A start job for unit nixos-upgrade.service has finished successfully.

   The job identifier is 3839.
Jan 23 04:40:53 shorthair systemd[1]: nixos-upgrade.service: Consumed 26.476s CPU time, 2.9G memory peak, 36K written to disk, 23.1K incoming IP traffic, 11.2K outgoing IP traffic.
   Subject: Resources consumed by unit runtime
   Defined-By: systemd
   Support: https://lists.freedesktop.org/mailman/listinfo/systemd-devel

   The unit nixos-upgrade.service completed and consumed the indicated resources.
```

Figure 2: Deploy to machine successful.