Repository Cleanup Execution Plan

HX-Infrastructure-Ansible Repository

Date: September 26, 2025
Repository: hanax-ai/HX-Infrastructure-Ansible
Current Branch Count: 34+ branches
Cleanup Objective: Streamline branch structure, implement governance, ensure security

---

Executive Summary

This repository has accumulated 34+ branches with inconsistent naming conventions and organizational issues. This plan provides a systematic approach to clean up the repository while maintaining operational safety and implementing proper governance.

Key Issues Identified:

- Multiple merge-fix branches (merge-fix-4, merge-fix-8, merge-fix-10, merge-fix-11, merge-fix-12)
- Temporary branches left unmerged (temp/workflow-validation)
- Copilot-generated branches (copilot/fix-*)
- Transfer work branches (transfer-work-*)
- Inconsistent feature branch naming
- No branch protection or governance policies

Phase 1A: Safety + Control Operations

1A.1 Safety Snapshot

- **Action:** Create pre-cleanup tag and bare repository backup
- **Tag:** `pre-cleanup-snapshot-YYYYMMDD`
- **Backup:** Full bare clone for rollback capability
- **Retention:** 90 days minimum

1A.2 Branch Protection Implementation

- **Target:** main branch (primary) + phase-1.0-deployment (current default)
- **Rules:**
- Require pull request reviews (1 reviewer minimum)
- Dismiss stale reviews when new commits are pushed
- Require status checks to pass before merging
- Restrict pushes to matching branches
- Allow force pushes: false
- Allow deletions: false

1A.3 Security Scanning

- **Tool:** gitleaks for sensitive data detection
- **Scope:** All branches and commit history
- **Output:** JSON report for review
- **Action:** Address any findings before proceeding

1A.4 Branch Inventory & Analysis

- **Data Collection:**
- Branch names and last commit dates
- Commit authors and activity patterns
- Merge status and relationships
- Staleness analysis (>30 days inactive)

1A.5 Governance Setup

- **CODEOWNERS:** Define team ownership and review requirements
- **Documentation:** Update README with branch policies
- **Workflow Hardening:** Implement required status checks

Branch Decision Matrix

DELETE (High Confidence)

Criteria: Temporary, merge artifacts, or abandoned work
- `merge-fix-4, merge-fix-8, merge-fix-10, merge-fix-11, merge-fix-12` - Merge artifacts
- `temp/workflow-validation` - Temporary testing branch
- `copilot/fix-*` - AI-generated fix attempts
- `transfer-work-*` - Work transfer branches (if merged)

CONSOLIDATE (Review Required)

Criteria: Related features that should be unified
- Phase/sprint feature branches with overlapping scope
- Multiple remediation branches addressing same issues
- Feature branches that can be rebased/squashed

KEEP (Active/Critical)

Criteria: Active development, deployment, or recovery branches
- `main` - Primary branch
- `phase-1.0-deployment` - Current default/active deployment
- Active feature branches with recent commits
- Recovery/backup branches with unique value

Risk Mitigation Strategies

1. Archive-Before-Delete

- Create archive tags for all branches before deletion

- Format: `archive/branch-name-YYYYMMDD`
- Retention: 6 months minimum

2. Rollback Capability

- Full bare repository backup before any changes
- Step-by-step operation logging
- Immediate rollback procedures documented

3. Stakeholder Communication

- Engineering team notification before execution
- 48-hour review period for branch decisions
- Clear escalation path for concerns

4. Gradual Execution

- Phase-based approach with validation checkpoints
- Ability to pause/resume operations
- Continuous monitoring of repository health

Implementation Timeline

Phase 1A: Safety + Control (Day 1)

- [x] Safety snapshot and backup creation
- [x] Branch protection implementation
- [x] Security scanning execution
- [x] Initial inventory and analysis
- [x] CODEOWNERS and governance setup

Phase 1B: Branch Decisions (Day 2)

- [ ] Stakeholder review of branch matrix
- [ ] Archive creation for deletion candidates
- [ ] Consolidation planning for related branches
- [ ] Final approval for cleanup operations

Phase 1C: Cleanup Execution (Day 3)

- [ ] Branch deletions (with archives)
- [ ] Branch consolidations and merges
- [ ] Final repository validation
- [ ] Documentation updates

Success Metrics

Quantitative

- Branch count reduction: Target 50%+ reduction (34+ → <17)
- Stale branch elimination: 100% of branches >90 days inactive

- Security scan: 0 critical findings
- Protection coverage: 100% of primary branches

Qualitative

- Clear branch naming conventions
- Documented ownership and policies
- Improved developer experience
- Reduced maintenance overhead

Rollback Procedures

Emergency Rollback

1. Restore from bare repository backup
2. Force push to restore branch state
3. Notify stakeholders of rollback
4. Investigate and document issues

Selective Rollback

1. Restore specific branches from archive tags
2. Recreate branch protection rules if needed
3. Update documentation to reflect changes
4. Communicate partial rollback to team

Post-Cleanup Governance

Branch Naming Convention

- `feature/description` - New features
- `fix/description` - Bug fixes
- `hotfix/description` - Critical production fixes
- `release/version` - Release preparation
- `docs/description` - Documentation updates

Branch Lifecycle

- Maximum lifetime: 30 days for feature branches
- Required reviews: 1+ for all merges to protected branches
- Automatic deletion: Merged branches after 7 days
- Stale branch alerts: Weekly notifications for >14 days inactive

Monitoring and Maintenance

- Weekly branch health reports
- Monthly governance policy reviews
- Quarterly cleanup assessments
- Annual policy updates

Plan Status: Phase 1A Ready for Execution
Next Action: Execute safety operations and create Phase 1A completion report