Engineering Spotlight:
ZTF-Orchestrator

A governed operations console for Nutanix automation

Community-built | ZTF 1.x operations | ZTF 2.x plan/apply lane | Controlled UAT boundary

Operations layer for ZeroTouch workflows

ZTF-Orchestrator turns ZeroTouch Framework, NKP preparation, approvals, durable jobs, validation evidence, and appliance operations into a browser-based control plane for infrastructure and field engineering teams.

Govern

RBAC, approval gates, audit history, and evidence capture before high-risk work.

Operate

Guided forms, YAML Studio, job queues, logs, schedules, and recovery-oriented runbooks.

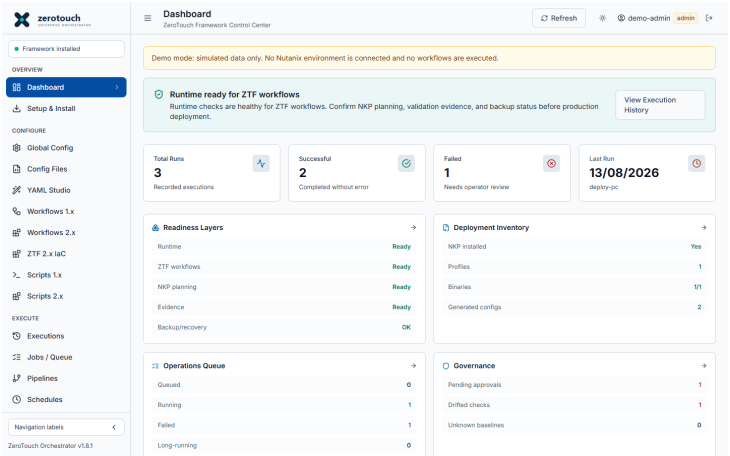
Validate

Clear separation between simulator, lab, controlled UAT, and production evidence.

ZTF-Orchestrator Feature Highlights

Configure - Approve - Execute - Evidence - Recover

- Browser operations console for ZTF 1.x workflows, scripts, and safe NKP phases.
- Separate ZTF 2.x plan/apply lane with approval-bound apply and destroy controls.
- Native Foundation Deploy intent planning with read-only evidence packs and gated UAT posture.
- PostgreSQL-backed appliance path plus local file-backed mode for simpler trials.



Key Features in ZTF-Orchestrator

- Guided workflow forms and YAML Studio expose the generated configuration instead of hiding it.
- Durable job queue captures submitted work, streamed logs, progress, cancellation, and history.
- Approval gates bind high-risk work to a review trail before execution.
- Validation Evidence creates downloadable, redacted handoff records for UAT and operations review.
- Drift detection, schedules, pipelines, and audit endpoints support repeatable team operations.
- Appliance Ops stages connected or air-gapped updates while keeping host-level changes explicit.
- Runtime compatibility protects legacy ZTF 1.x execution from being launched through the wrong ZTF 2.x CLI.

Operating Nutanix Automation at Team Scale

Workflow Control

Allowlisted ZTF 1.x workflows, dry-run checks, guarded submit, and execution history.

Governance

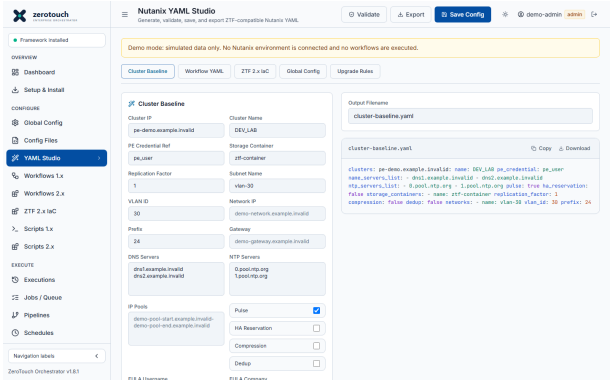
RBAC, approvals, audit records, schedules, and operator evidence requirements.

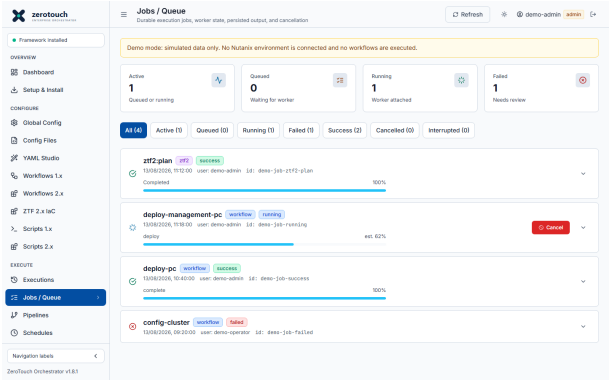
ZTF 2.x Readiness

Plan submissions, hash-bound approvals, and guarded apply/destroy foundations.

Appliance Operations

Docker, PostgreSQL, air-gapped updates, backup, rollback, and health guidance.





The Operator Challenge

Repeatability

Teams need consistent cluster, PC, PE, NKP, and baseline workflows without every run becoming a bespoke CLI exercise.

Governance

High-impact infrastructure automation needs approval, RBAC, auditability, and recovery paths before it touches targets.

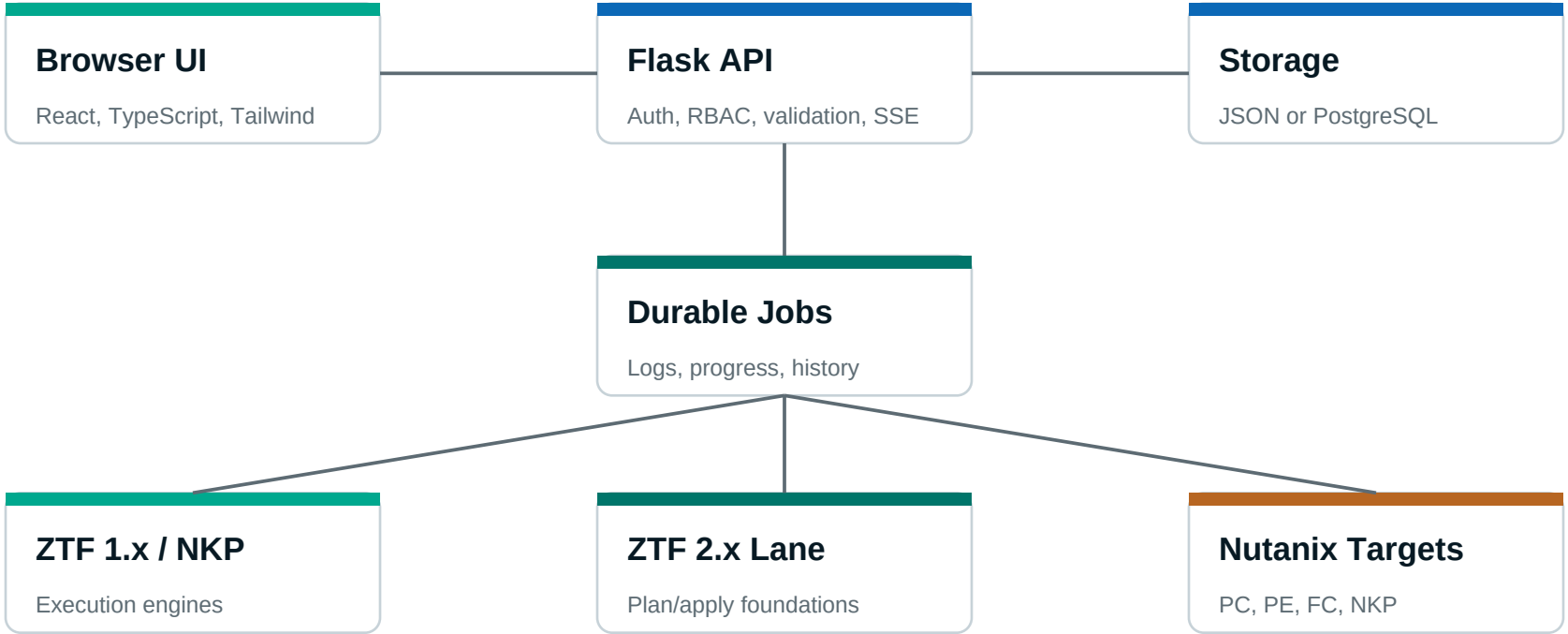
Evidence

Local demos, lab runs, controlled UAT, and production validation must stay distinct and traceable.

Architecture and Function

Browser workflow, API controls, durable jobs, storage, and automation adapters

Basic ZTF-Orchestrator Architecture



How Teams Run ZTF-Orchestrator

Local Trial

Manual install or one-command script for quick operator review and development.

Team Service

Docker Compose with PostgreSQL for durable state, audit, jobs, backups, and appliance flows.

Appliance / AHV

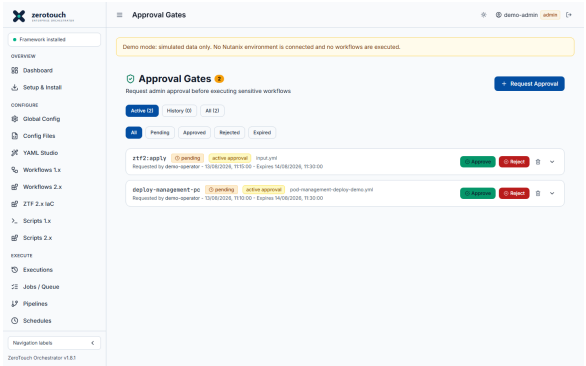
VM-oriented deployment kit, first boot setup, update staging, and rollback runbooks.

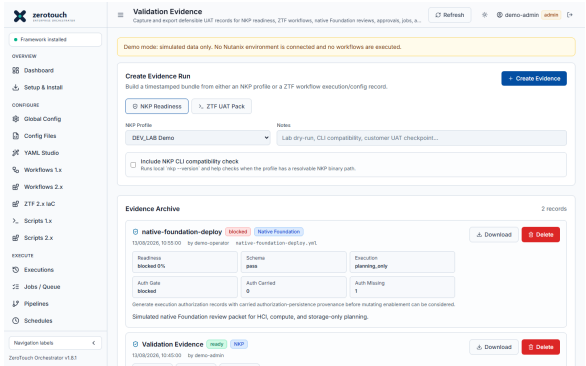
- Static demo is available for UI walkthroughs with simulated data only.
- Connected and air-gapped update paths are separate from QCOW2 image replacement.
- External exposure requires reverse proxy, TLS, and environment hardening.

ZTF-Orchestrator Workflow

1. Configure runtime paths, credentials, profiles, and deployment mode.
2. Generate or import YAML through guided forms and YAML Studio.
3. Validate compatibility, dry-run behavior, and readiness gates.
4. Request approval for governed workflows or high-risk phases.
5. Submit durable jobs and watch streamed logs, progress, and history.
6. Capture validation evidence and recovery notes for UAT handoff.

Governed Workflow and Evidence Model





- Engineers submit workflow configuration and review generated YAML before execution.
- Approvers review request metadata, risk, workflow ID, plan bindings, and required evidence.
- Jobs persist operator-visible logs, terminal state, diagnostics, and downloadable evidence packs.
- Production claims remain scope-specific and require target-side validation evidence.

Thank You

Portfolio: https://www.johngoulden.de/

Public repository: https://github.com/VirtuArchitect/ZTF-Orchestrator

Static demo: https://virtuarchitect.github.io/ZTF-Orchestrator/

Demo video: raw.githubusercontent.com/VirtuArchitect/ZTF-Orchestrator/main/docs/assets/video/ztf-orchestrator-product-demo-90s-pro.mp4

Boundary: Independent community project. Not affiliated with or supported by Nutanix.