

Hoarding node Live Network Deployment

Live Network Deployment

High-level Description

The hoarding node has been developed and validated in local testing environments. To collect real observability data from the Cardano network, it needs to run continuously against a live network — connected to real peers, exposed to real traffic, and accumulating data over time. This work package covers the deployment infrastructure and live operation of the hoarding node against the Cardano pre-production network. It includes the full deployment pipeline — AWS infrastructure provisioned as code, automated deployments from CI, encrypted secrets management, and automated database migrations — as well as six months of live operation on preprod.

The deployment is built entirely from infrastructure-as-code and serves a dual purpose: it is the team's own live environment for collecting real data, and it is the reference implementation that external operators can follow to run their own instance. External operators need only the NixOS module or OCI image and a way to provide secrets as environment variables — no separate infrastructure repository or tooling required.

The distributed mode and a mainnet deployment are explicitly out of scope for this proposal.

Deliverables:

- Terraform, NixOS configuration, and CI/CD pipeline for the team's preprod deployment.
 - `nixosModules.hoard` flake export and OCI image, documented for external operators.
 - Six months of live operation against preprod, with observability dashboards.
-

Core Objectives

- First live deployment on a real Cardano network — the hoarding node moves from local testing to continuous operation against pre-production, collecting real blocks, headers, and peer data over an extended period.
- Fully automated, reproducible deployments — infrastructure is defined as code and deployments are triggered automatically from CI. No manual steps are required after initial setup; any team member can deploy.
- Low barrier to entry for external operators — any NixOS or Docker operator can run their own hoarding node instance by importing the flake module or running the OCI image, with secrets provided as environment variables.

- Operational visibility — observability dashboards provide continuous insight into collection activity, peer connectivity, and system health, enabling the team to detect and respond to issues promptly.
-

Expected Value

The hoarding node's purpose is to observe the Cardano network and surface anomalies. Without a live deployment, that purpose is theoretical — local testing can validate correctness but cannot produce the sustained, real-world data that makes the tool valuable.

This work package is the enabling condition for the rest of the project. The validation work package's success criterion of detecting adversarial behaviour in the wild, the transaction collection work package's inbound connection fallback path, and the distributed mode's multi-region observation goals all require a live environment to be meaningful. This proposal creates that environment.

Six months of operation against pre-production provides enough time to accumulate a dataset worth analysing: orphaned blocks, classification patterns, peer behaviour, and — once the validation work package is delivered — consensus-level invalidity. The observability stack makes this data continuously visible rather than requiring manual inspection.

Making deployment accessible to external operators compounds this value. Each additional operator running the hoarding node adds an independent vantage point on the network. The NixOS module and OCI image lower the barrier enough that any operator already running a Cardano node can also run a hoarding node alongside it.

Metrics for Success

- Hoarding node running continuously against pre-production, with observability dashboards operational.
 - Real orphaned/invalid block data being collected and queryable via the HTTP API.
 - At least one publicly exposed report of adversarial behaviour detected and evidenced using data collected by the hoarding node.
 - If the transaction collection proposal is accepted and connection promotion proves insufficient: transactions being collected via inbound connections in the same environment.
-

Strategic Alignment

Pillar: Pillar 1 — Infrastructure & Research Excellence (Focus Area I.2: Threat Detection & Recovery)

Rationale: A live deployment is the prerequisite for all network observability work. Without a node running continuously against a real network, no adversarial behaviour can be detected or documented. The infrastructure-as-code pipeline also contributes to the operational reliability goals of Focus Area I.2.

KPI support: The existing Pillar 1 KPIs are not direct outputs of this proposal, but a live deployment is the foundation for demonstrating all other proposals' impact on network health.

Classification

- New initiative or continuation of existing: Continuation
 - Primary nature: Technical
-

Milestones

Milestone 1 — Deployment Pipeline

Deliverables: AWS resources provisioned via Terraform (EC2, RDS, VPC, security groups). NixOS configuration defined in the flake. One-time bootstrap via nixos-anywhere installs NixOS on the EC2 instance from the flake — no manual machine setup required. Automated CI/CD pipeline deploying to the EC2 instance via deploy-rs on every push to main. Secrets managed via sops-nix. Database migrations automated via sqitch, running before the application starts on each deploy, with automatic rollback on failure.

Acceptance criteria: A push to main triggers a fully automated deployment to a running EC2 instance with RDS. Secrets are never stored in plaintext. A failed migration aborts and rolls back the deploy automatically.

Milestone 2 — Live Preprod Deployment

Deliverables: Hoarding node running against the Cardano pre-production network. Full observability stack operational: Prometheus and Grafana for metrics and dashboards, Loki and Promtail for log aggregation, Tempo for distributed tracing, and node_exporter for host-level metrics. Dashboards showing live block and header collection, peer connectivity, and system health metrics.

Acceptance criteria: Hoarding node is running and collecting data against preprod. Orphaned and invalid block data is queryable via the HTTP API. Observability dashboards are operational and alerting is configured.

Milestone 3 — External Operator Artifacts

Deliverables: nixosModules.hoard exported from the flake, with configurable options and secrets injected via environment variables. OCI image documented as a secondary deployment path for non-NixOS operators. The team's own preprod deployment serves as the reference implementation.

Acceptance criteria: An external operator can deploy the hoarding node by importing the NixOS module or running the OCI image, following only the published documentation.

Note: a Cardano node is currently a required shared component for the hoarding node to function. This dependency is removed if the embedded consensus validation work package is approved.

Supporting Material

- <https://github.com/tweag/cardano-hoarding-node/blob/main/proposal/live-network-deployment/proposal.md> — budget ask, cloud cost estimate, deliverables, success criteria
- <https://github.com/tweag/cardano-hoarding-node/blob/main/proposal/live-network-deployment/overview.md> — deployment modes and shared components
- <https://github.com/tweag/cardano-hoarding-node/blob/main/proposal/live-network-deployment/single-node.md> — full single-node stack: Terraform, NixOS, deploy-rs, sops-nix, sqitch, CI/CD