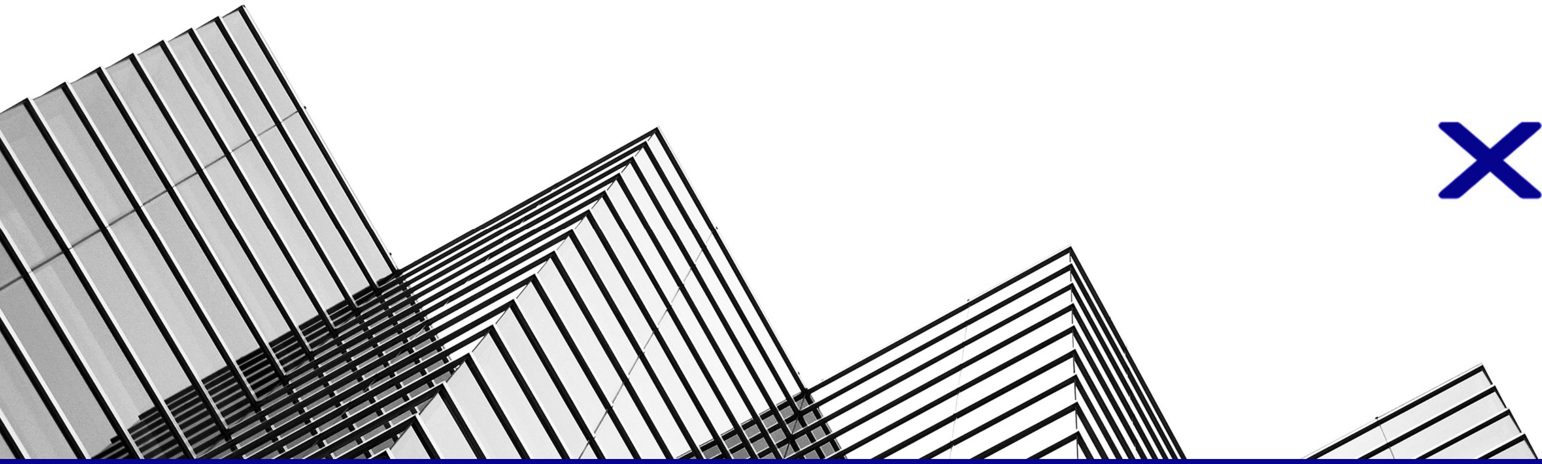
Building an Open-Source AI Factory with Upstream Projects - A Primer



XAASIO

Open-Source AI Factory

Why an Open Source AI Factory?

XAASIO

Avoid lock-in to any single vendor or proprietary AI control plane

Run the same stack on:

- On-prem / OpenStack
- Public cloud
- Edge & remote sites

Reuse and contribute to **upstream projects** instead of forks

Give enterprises and governments:

- Transparency (source code)
- Sovereignty (run where you want)
- Extensibility (swap components as needed)

Conceptual Architecture

Application Layer
Digital Twin, Code Gen, Computer Vision, Agentic AI, Enterprise RAG, Virtual Assistants

Application Platform Layer
“Everywhere Inference”, notebooks, pipelines, model registry, feature store

Platform Layer
Kubernetes clusters on bare metal, OpenStack, or public cloud

Compute Layer
OS, container runtime, GPU drivers & device plugins

Infrastructure Layer
Compute, GPU, network, storage hardware

Management & Observability
Fleet lifecycle, GitOps, observability, service catalog

Infrastructure & Compute Layer (Upstream)

Infrastructure

- Commodity x86/ARM servers with NVIDIA / AMD / Intel GPUs
- High-speed L3 fabric (leaf–spine) with VLAN/VXLAN segmentation
- Storage:
  - Ceph for block, file, and object
  - Ceph RGW for S3-compatible object store

Compute Layer

- OS: Ubuntu / Debian / Rocky / AlmaLinux etc.
- Container runtime: containerd or CRI-O
- GPU drivers from vendors + Kubernetes device plugins
- Optional: MIG/SR-IOV where available for GPU/NET slicing

Platform Layer: Kubernetes Everywhere

Kubernetes as a Service using upstream tooling:

- Cluster API (CAPI) for multi-cluster lifecycle
- Metal3 / Ironic for bare-metal provisioning
- Kubespray or kOps for simple cluster bootstrap

Multi-environment support:

- Bare metal / OpenStack clusters
- Public cloud K8s (self-managed or managed)
- Edge clusters with K3s or MicroK8s

Standard CNI plugins: Cilium or Calico for pod networking and Network Policies

Data Lake & Lakehouse

Object & block storage: Ceph (S3 API)

Lakehouse formats:

- Apache Iceberg / Apache Hudi / Delta Lake on S3 or Ceph RGW

Batch & distributed compute:

- Apache Spark & Apache Flink for ETL, feature generation, offline training data

Benefits:

- ACID tables on cheap object storage
- Time-travel & schema evolution for ML workloads
- Shared foundation for BI and ML

Data Ingestion, Streaming & ETL

Streaming backbone: Apache Kafka (or compatible like Redpanda)

Ingestion & CDC:

- Apache NiFi for pipelines from files, APIs, DBs
- Debezium for change-data-capture from OLTP DBs

Workflow & ETL orchestration:

- Apache Airflow or Argo Workflows for data pipelines

Outcome: unified, streaming-friendly data platform feeding the AI Factory

Feature Store, Analytics & Metadata

Feature Store: Feast, backed by Kafka + online store (Redis/Postgres)

Experiment & lineage metadata:

- MLflow tracking for runs, params, metrics, artifacts
- OpenLineage + Marquez (or Kubeflow Metadata) for data & pipeline lineage

Analytics:

- Apache Superset or Metabase for dashboards over the lakehouse

Enables consistent, shareable, and auditable features across teams

IAM, SSO & Directory (Keycloak + FreeIPA)

FreeIPA

- Central directory: users, groups, host identities, Kerberos, LDAP, DNS

Keycloak

- Identity provider: OpenID Connect, OAuth2, SAML
- Multi-tenant realms; role-based access control (RBAC)

Integration pattern

- FreeIPA as the system of record
- Keycloak uses FreeIPA as user/group source
- All platform UIs & APIs (Cloudscape console, JupyterHub, Kubeflow, MLflow, Grafana,
- KServe endpoints) use Keycloak for login & tokens
- Result: enterprise-grade SSO and authorization across the AI Factory

Experimentation & Pipelines (MLOps Core)

Notebooks & IDEs

- JupyterHub / Kubeflow Notebooks with GPU-enabled profiles

Pipelines & training orchestration

- Kubeflow Pipelines for ML workflows on Kubernetes
- Airflow or Argo Workflows for data + ML hybrid workflows

Distributed training

- Ray, Horovod, DeepSpeed, or Torch DDP for multi-GPU/multi-node training

All runs log metrics & artifacts into MLflow for comparison and governance

Model Registry, CI/CD & Promotion

Model registry: MLflow Model Registry (or similar)

GitOps CI/CD:

- Argo CD or Flux CD manages Kubernetes manifests for serving stacks
- Tekton or GitHub/GitLab CI for build-and-test pipelines

Promotion workflow:

- Data scientist promotes a model version in MLflow →
- CI pipeline generates KServe or deployment manifests →
- Git commit → Argo/Flux deploys to staging → then production after approval

Gives clear versioning, approvals, and rollback for models

Online Serving & “Everywhere Inference”

Serving frameworks:

- KServe for standardized model inference (REST/gRPC)
- NVIDIA Triton or TorchServe for DL workloads behind KServe or standalone

LLM & GenAI serving:

- vLLM or Text Generation Inference (TGI) for high-throughput LLMs
- Ray Serve for agentic or multi-model apps

Everywhere inference pattern:

- Same serving stack deployable to:
  - Core DC clusters
  - Edge K3s/MicroK8s clusters
  - Cloud clusters

Rollout via GitOps, using Keycloak auth and shared telemetry

Observability & FinOps

Metrics & alerts: Prometheus + Alertmanager

Dashboards: Grafana (infra, K8s, apps, models, business KPIs)

Logs & traces:

- Loki for logs
- Tempo or Jaeger + OpenTelemetry for traces

Cost & utilization:

- OpenCost (or cost exporter) for per-namespace / per-workload cost visibility
- Observability data surfaces in the Cloudscape console for operators and tenants

Security, Policy & Supply-Chain

Policy-as-code:

- OPA Gatekeeper or Kyverno for admission policies (allowed images, labels, limits, namespaces)

Secrets & KMS:

- HashiCorp Vault or KMS-encrypted Kubernetes secrets

Runtime security:

- Falco or similar for syscall-level anomaly detection

Software supply chain:

- Container image signing with Sigstore Cosign
- Harbor (or other OCI registry) for scanning and provenance
- Combined with Keycloak/FreeIPA, this yields a secure multi-tenant AI Factory

End-to-End Application Flow

Onboard & authenticate

- Admin creates a tenant/project in Cloudscape; users log in via Keycloak (backed by FreeIPA).

Create workspace

- Workspace maps to K8s namespaces, quotas, and policies via Gatekeeper/Kyverno.

Ingest & prepare data

- Data flows via NiFi / Kafka / Debezium into Ceph; ETL with Spark/Flink.

Engineer features

- Data scientists use Jupyter/Kubeflow Notebooks, store features in Feast and lakehouse tables.

Train & experiment

- Pipelines orchestrated by Kubeflow or Airflow; distributed training with Ray/Horovod; metrics logged to MLflow.

Register & approve model

- Best run promoted in MLflow registry; CI pipeline generates deployment manifests.

Deploy & serve

- Argo/Flux deploy KServe/vLLM/Triton endpoints; Ingress/mesh exposes secured APIs.

Monitor, optimize, retrain

- Prometheus/Grafana/OpenCost + Evidently AI track performance, cost, drift → trigger retraining cycles.

XAASIO

With Endless Curiosity and Dedicated Passion. Thank You