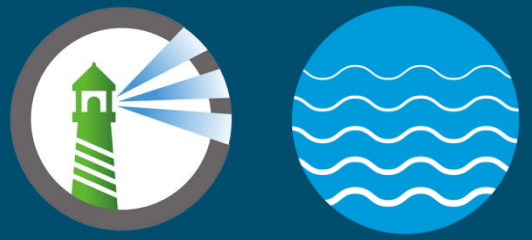
KubeCon CloudNativeCon India 2026

Cloud Native AI 

Model Management with Harbor & Velero

KubeCon India 2026

Dhruv Tyagi · Product Marketing Engineer, Broadcom

THE PROBLEM

LLMs don't fit in container images

1

Size

Multi-gigabyte models bloat container images. TinyLlama 1.1B is 668 MB. GPT-scale models exceed 70 GB.

2

Coupling

Updating the model requires rebuilding and redeploying the entire image. Inference code and weights ship as one frozen artifact.

3

Ops gap

No versioning, no deduplication, no metadata. Models are treated as opaque binary blobs.

THE SHIFT

The old way bakes models into images. The cloud native way doesn't.

✕ The old way	✓ The cloud native way
Model baked into container image	Model stored in OCI registry
No versioning	Semantic versions + rich metadata
Re-pull entire image on update	Layer deduplication — pull only the delta
Model tightly coupled to code	Model and inference server decouple
No Annotations	Annotations: architecture, format, params

THE SPEC

OCI wasn't just for containers — it was always for artifacts

OCI defines a spec for storing **any artifact** in a registry. Containers were the first use case, not the only one.

CNAI (Cloud Native AI) extends OCI with a standard set of annotations for AI models — a shared vocabulary so any registry, any tool, any platform speaks the same language about models.

ORAS — OCI Registry As Storage — is the CLI for pushing and pulling non-container OCI artifacts.

CNAI annotations

org.cnai.model.format

org.cnai.model.architecture

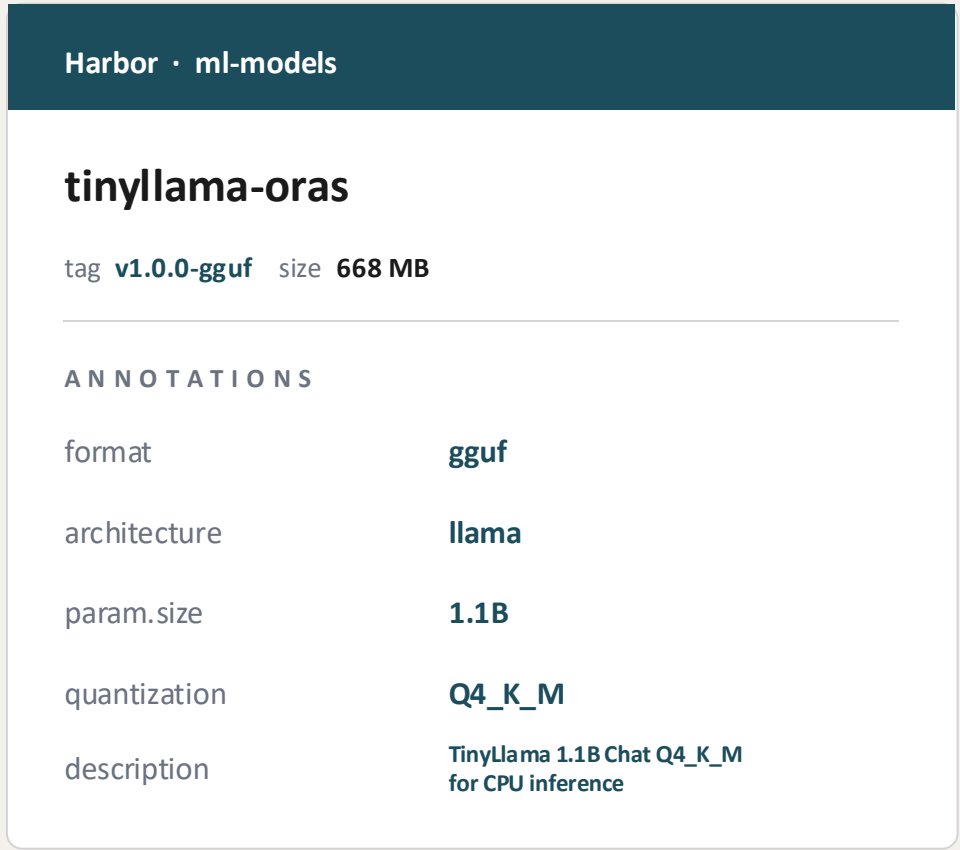
org.cnai.model.param.size

org.cnai.model.quantization

Your registry becomes a first-class model store — same tools, same RBAC, same replication.

Harbor 2.13 — your registry already supports this

- Harbor is **CNCF-graduated** and **OCI-compliant** out of the box
- In **Harbor 2.13+**, CNAI artifacts surface in the UI with their metadata, automatically
- Platform teams get RBAC, vulnerability scanning, replication policies, and quota — **all applied to models**, not just containers
- **One registry** for containers, Helm charts, and AI models



From Hugging Face to inference in 4 steps

1

HUGGING FACE

Download

Pull quantized GGUF model model from Hugging Face Hub Hub

2

Push

Push to Harbor as an OCI artifact with CNAI annotations using oras push — Harbor stores the model with full metadata, versioned and deduplicated

3

Deploy

Kubernetes init container pulls model from Harbor into a shared PVC; inference server starts

4

Serve

llama.cpp server exposes /v1/chat/completions; application sends queries

The inference server image never changes when the model updates — only the OCI tag in Harbor does.

You backed up your manifests. Did you back up the AI model?

Namespace deleted accidentally

Model cache PVC gone. Cold restart re-pulls from Harbor (or worse, Hugging Face). Minutes of downtime per replica.

Cluster migration

Stateful AI workloads don't come with you.

Compliance / audit

No point-in-time snapshot of model + config together.

Standard Velero backs up Kubernetes resources. Add Kopia file-system backup and it captures PVC contents too — including your model cache.

THE FIX

One command backs up your entire AI workload

```
$ velero backup create ai-backup --include-namespaces ai-inference
```

WHAT GETS CAPTURED

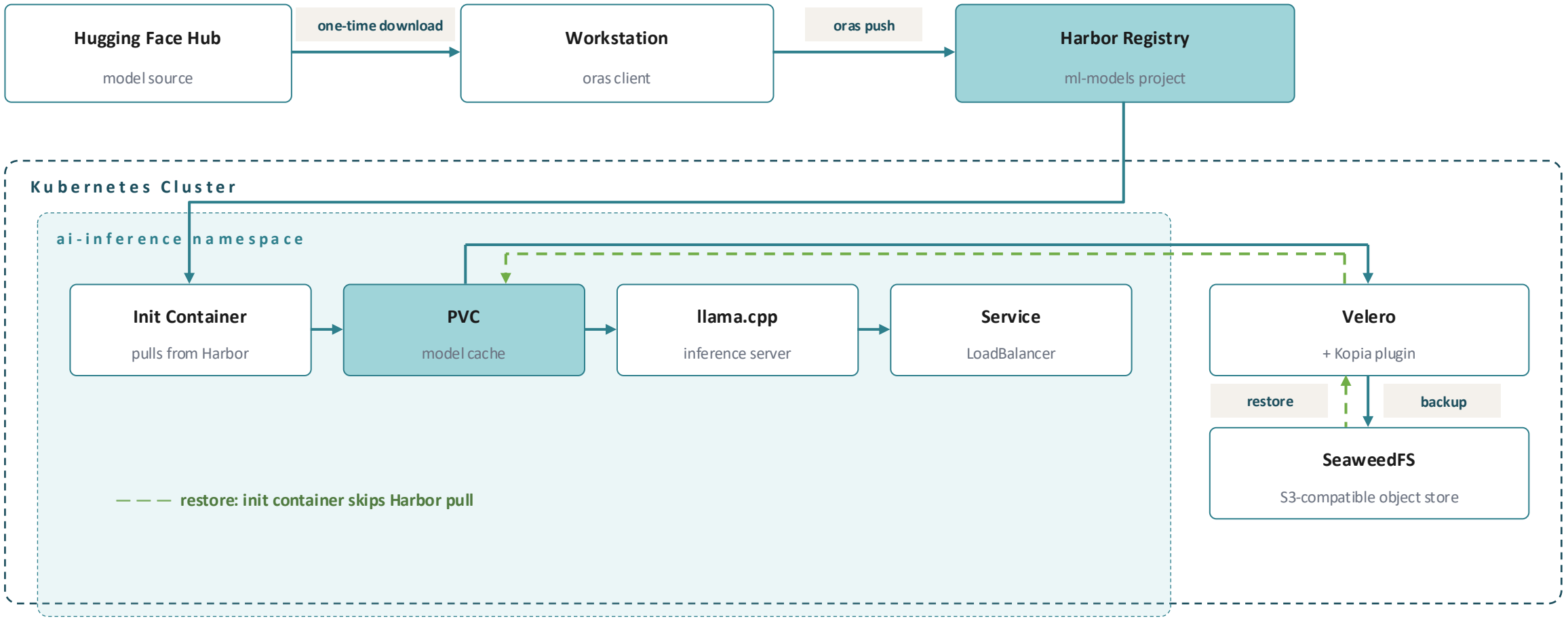
- ✓ Deployment, Service, ConfigMap (inference server config)
- ✓ Secrets (Harbor credentials)
- ✓ PersistentVolumeClaim + contents (668 MB model file via Kopia) Kopia)

WHAT HAPPENS ON RESTORE

- ✓ All resources recreated in the cluster
- ✓ PVC restored with model already present
- ✓ Init container detects cache hit → skips Harbor pull entirely

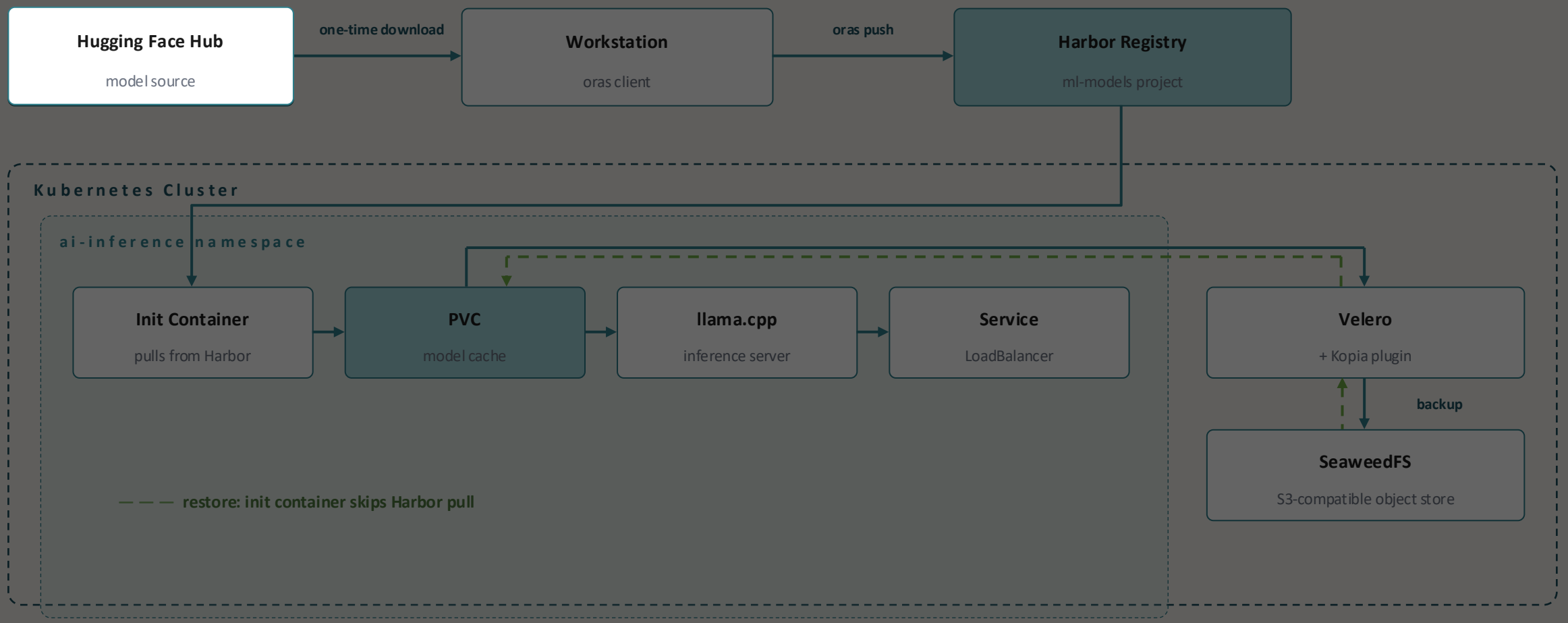
Recovery time drops from minutes (re-download) to seconds (cache restored).

The full architecture, end to end



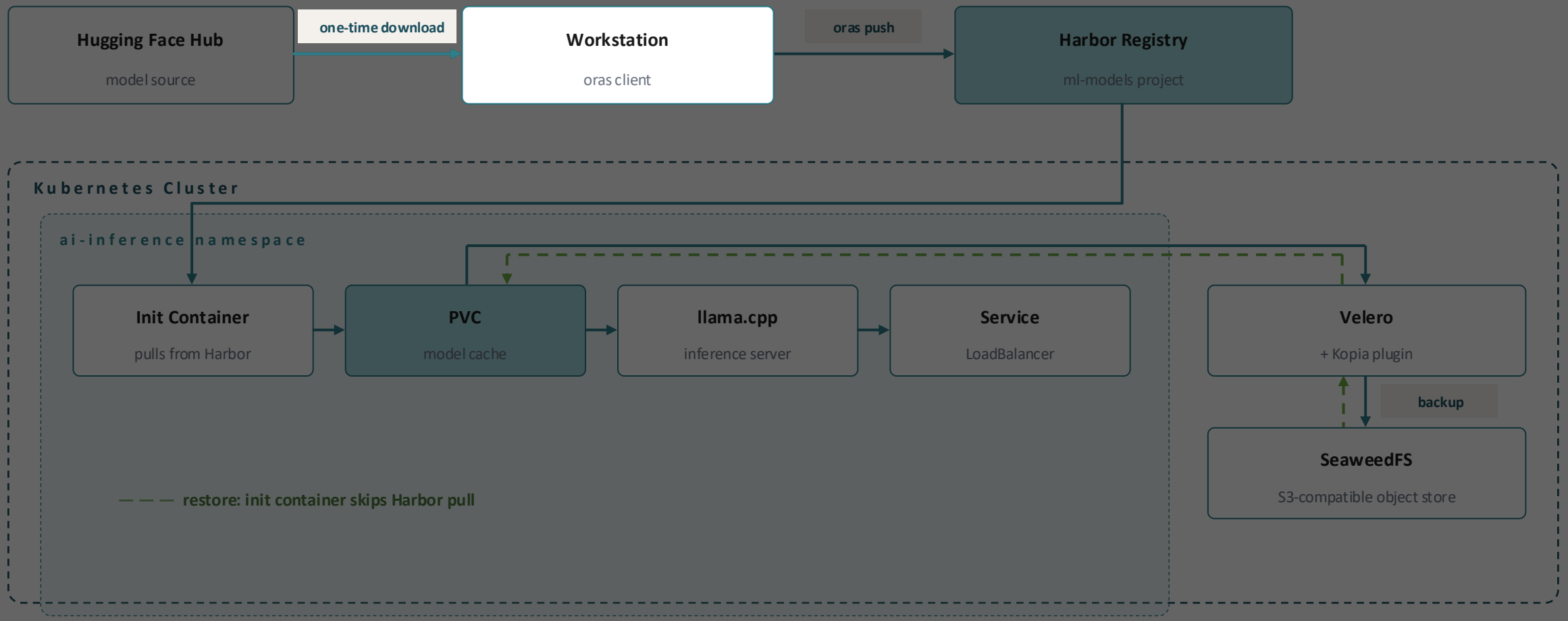
Harbor 2.14.2 · ORAS CLI · llama.cpp · TinyLlama 1.1B Q4_K_M (GGUF) · Velero 1.18.1 + Kopia · SeaweedFS 3.73 · Kubernetes 1.35

The full architecture, end to end



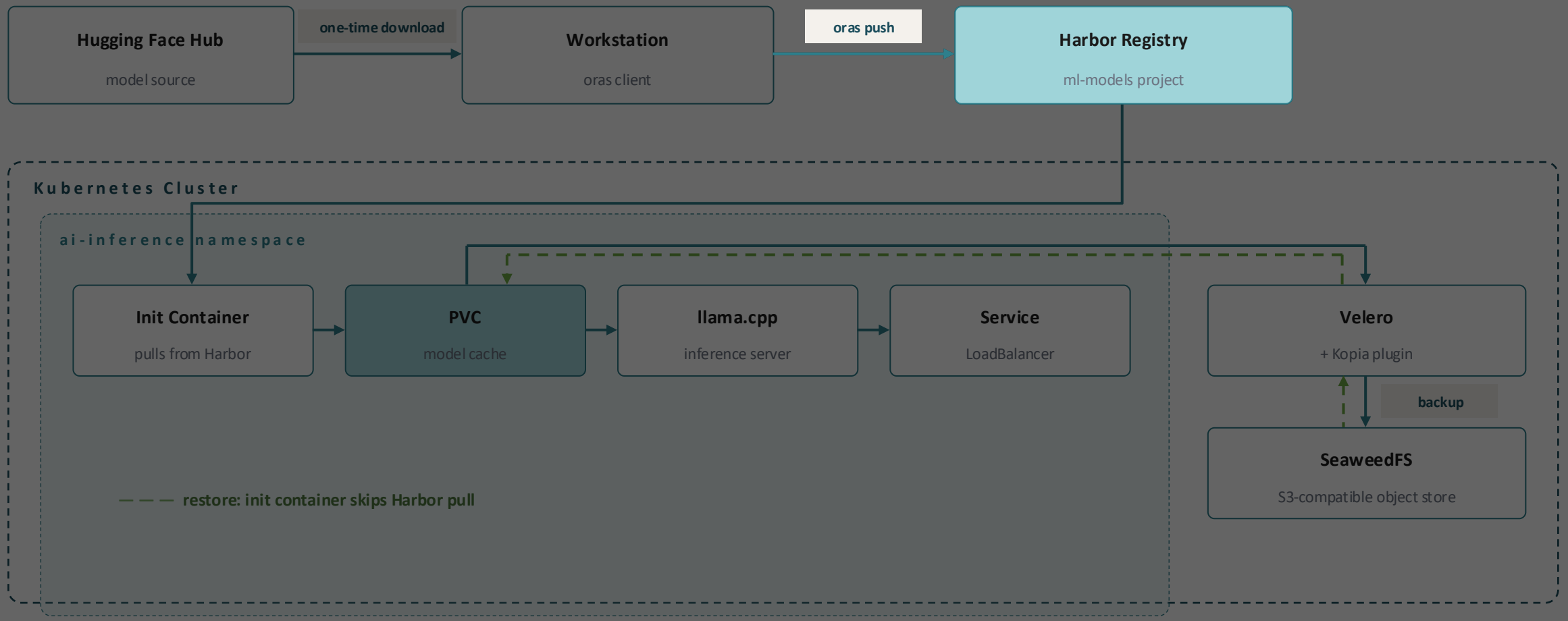
Harbor 2.14.2 · ORAS CLI · llama.cpp · TinyLlama 1.1B Q4_K_M (GGUF) · Velero 1.18.1 + Kopia · SeaweedFS 3.73 · Kubernetes 1.35

The full architecture, end to end



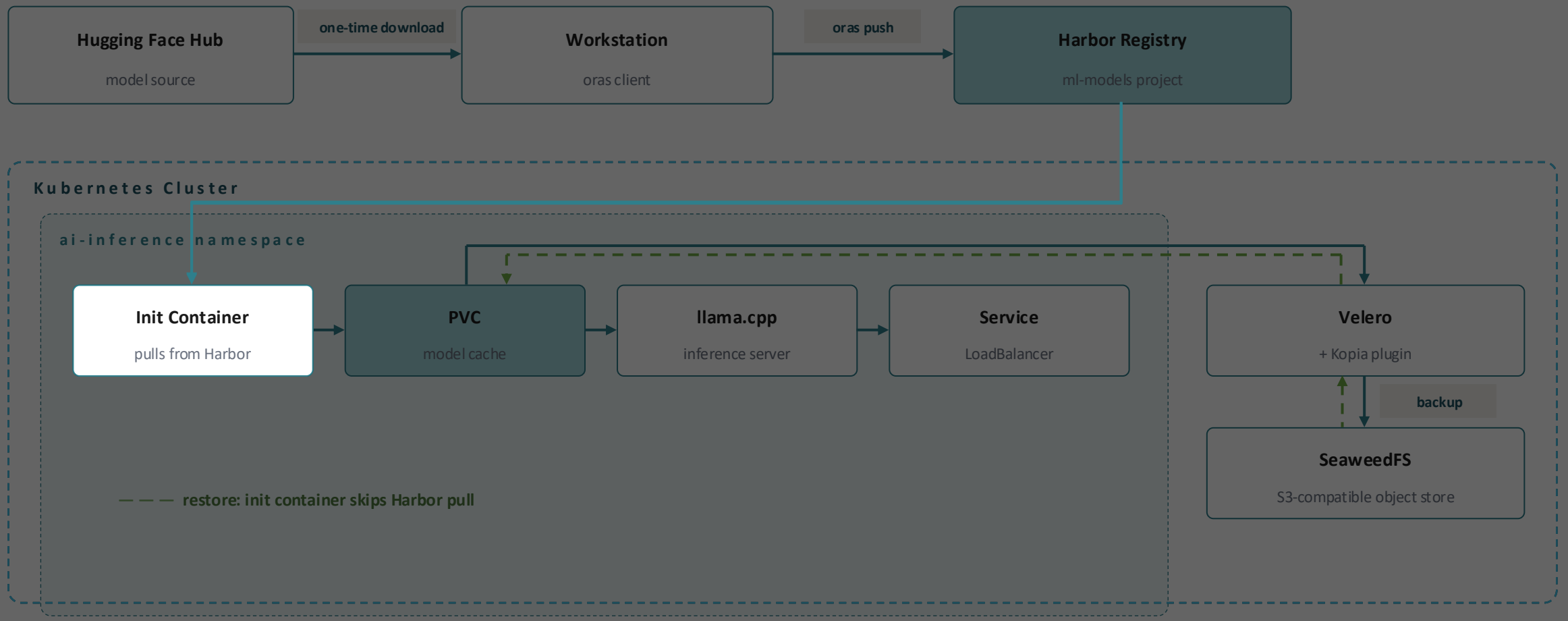
Harbor 2.14.2 · ORAS CLI · llama.cpp · TinyLlama 1.1B Q4_K_M (GGUF) · Velero 1.18.1 + Kopia · SeaweedFS 3.73 · Kubernetes 1.35

The full architecture, end to end



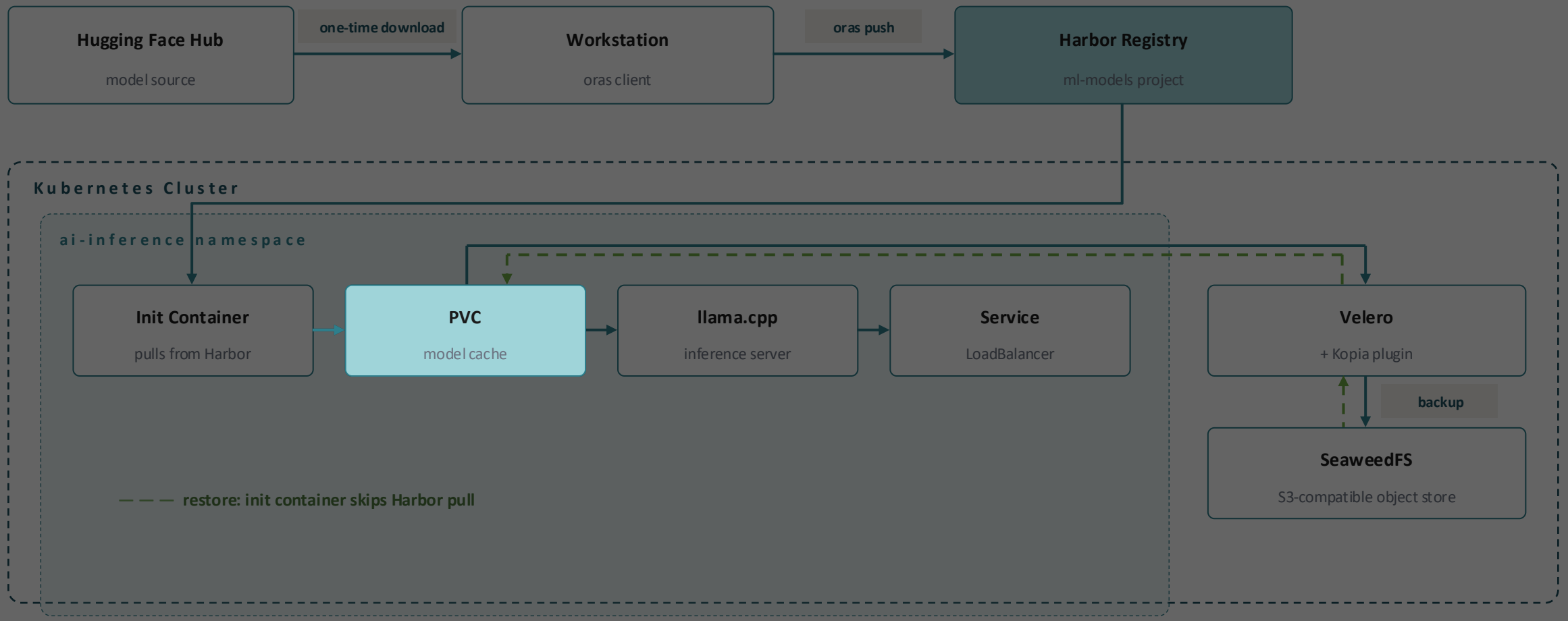
Harbor 2.14.2 · ORAS CLI · llama.cpp · TinyLlama 1.1B Q4_K_M (GGUF) · Velero 1.18.1 + Kopia · SeaweedFS 3.73 · Kubernetes 1.35

The full architecture, end to end



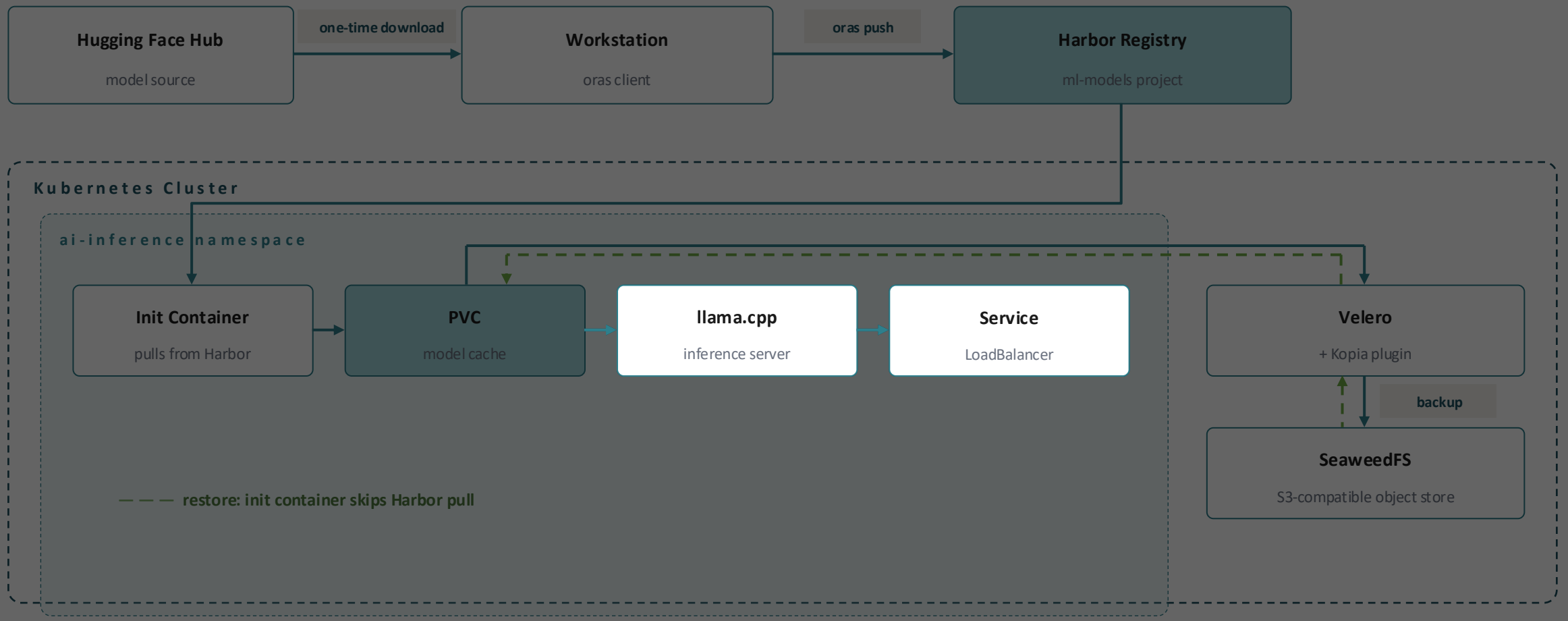
Harbor 2.14.2 · ORAS CLI · llama.cpp · TinyLlama 1.1B Q4_K_M (GGUF) · Velero 1.18.1 + Kopia · SeaweedFS 3.73 · Kubernetes 1.35

The full architecture, end to end



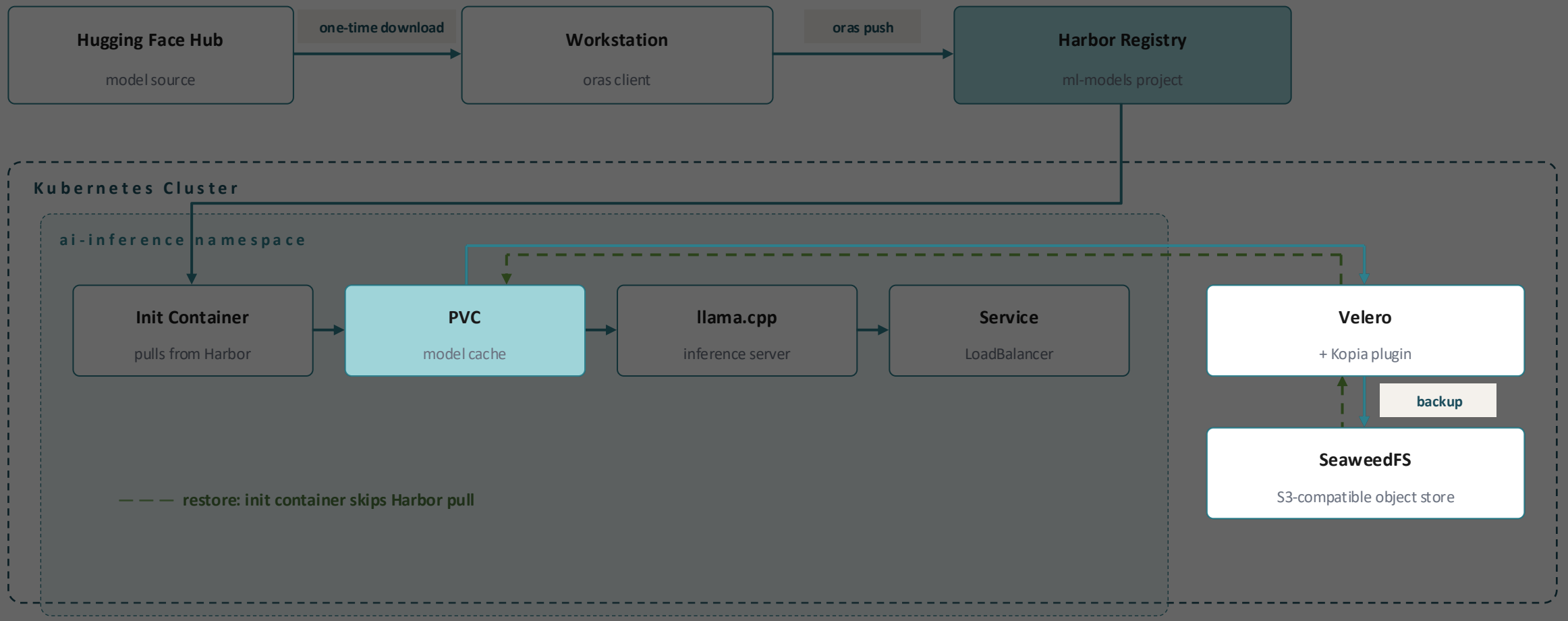
Harbor 2.14.2 · ORAS CLI · llama.cpp · TinyLlama 1.1B Q4_K_M (GGUF) · Velero 1.18.1 + Kopia · SeaweedFS 3.73 · Kubernetes 1.35

The full architecture, end to end



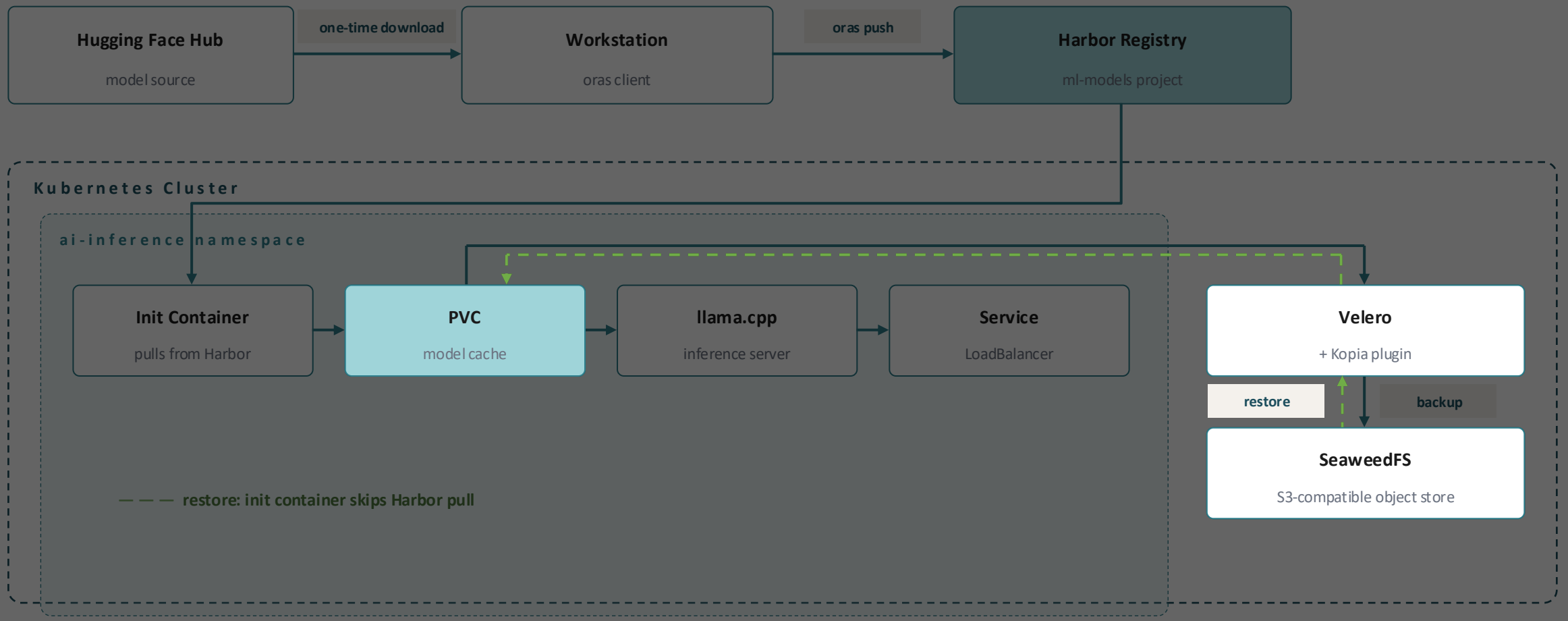
Harbor 2.14.2 · ORAS CLI · llama.cpp · TinyLlama 1.1B Q4_K_M (GGUF) · Velero 1.18.1 + Kopia · SeaweedFS 3.73 · Kubernetes 1.35

The full architecture, end to end



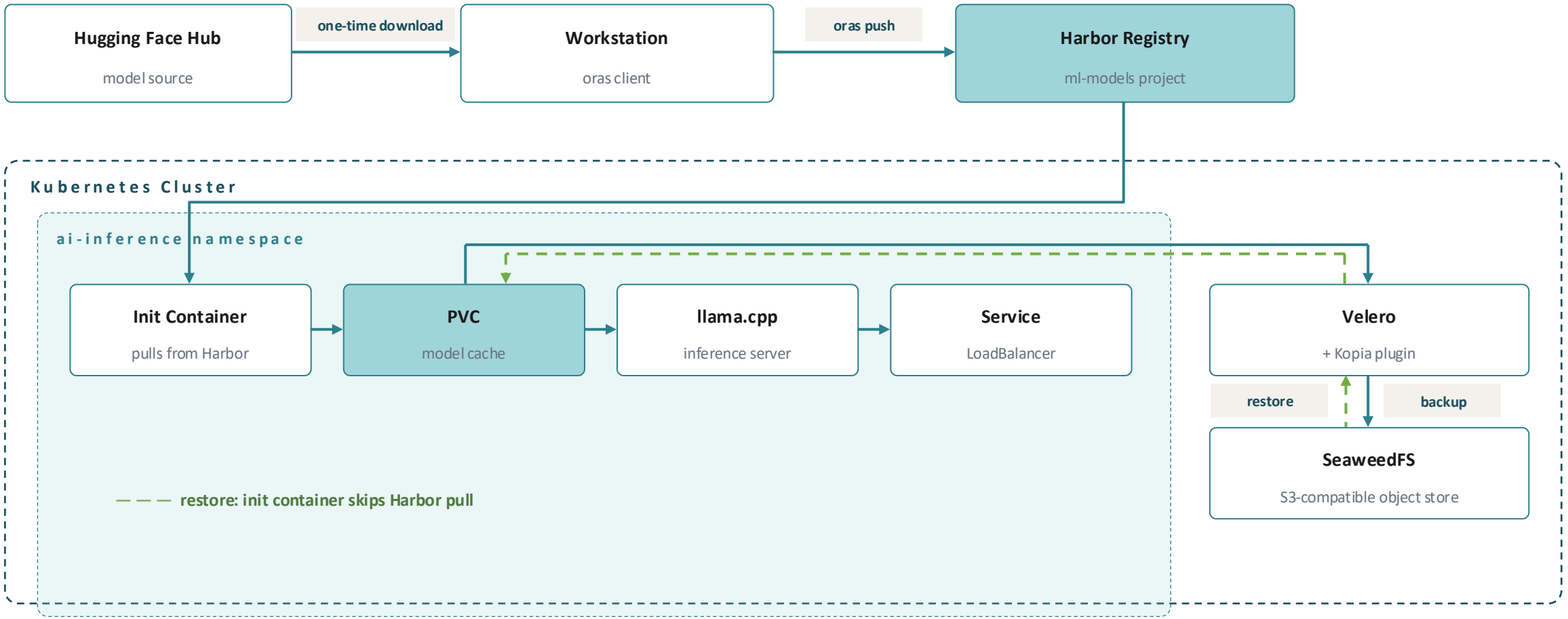
Harbor 2.14.2 · ORAS CLI · llama.cpp · TinyLlama 1.1B Q4_K_M (GGUF) · Velero 1.18.1 + Kopia · SeaweedFS 3.73 · Kubernetes 1.35

The full architecture, end to end



Harbor 2.14.2 · ORAS CLI · llama.cpp · TinyLlama 1.1B Q4_K_M (GGUF) · Velero 1.18.1 + Kopia · SeaweedFS 3.73 · Kubernetes 1.35

The full architecture, end to end



Harbor 2.14.2 · ORAS CLI · llama.cpp · TinyLlama 1.1B Q4_K_M (GGUF) · Velero 1.18.1 + Kopia · SeaweedFS 3.73 · Kubernetes 1.35

DEMO TIME

Let's see it live

01 Push TinyLlama to Harbor as a CNAI OCI artifact

02 Deploy llama.cpp inference server — init container pulls model from Harbor

03 Velero backup + full namespace restore — restored pod skips Harbor pull

THE TAKEAWAYS

Three things to take home

1

Your OCI registry is already a model registry.

Harbor + CNAI turns a standard registry into a versioned, metadata-rich model store — with zero additional infrastructure.

2

Decouple your model from your inference server.

Update, roll back, or A/B test models by changing an OCI tag — not rebuilding containers.

3

Velero + Kopia = full AI namespace DR.

One backup command captures manifests, configs, secrets, and the model cache PVC together as an atomic snapshot.

Harbor is your AI Model Registry

Velero is your backup provider

Dhruv Tyagi

Product Marketing Engineer, Broadcom

Resources

- github.com/goharbor/harbor
- github.com/velero-io/velero

linkedin.com/in/dhruv-tyagi-2015