Infrastructure

What is InferenceOps?

Learn what InferenceOps is, why it matters, and how leading AI teams scale, optimize, and manage LLM inference for production-grade performance and reliability.

You built an LLM-powered application. The demo went great. Leadership is impressed. Applause. Green lights. Your team is ready to ship.

But fast forward a few weeks:

- Your GPU usage is climbing fast. Costs are spiking. How do you secure enough compute, optimize cost, and ensure reliability?
- You are exploring different distributed inference strategies, but can't tell which parallelism strategy or inference setup best fits your use case.
- A new frontier open model just dropped. Can you immediately launch a deployment and start experimenting or are you stuck waiting on infrastructure support?
- A model misfires in production. Can you trace, debug, and fix it, without slowing your team down?

These aren't edge cases. They're the new normal when inference scales. The problem isn't just missing tools. It's a deeper operational gap.

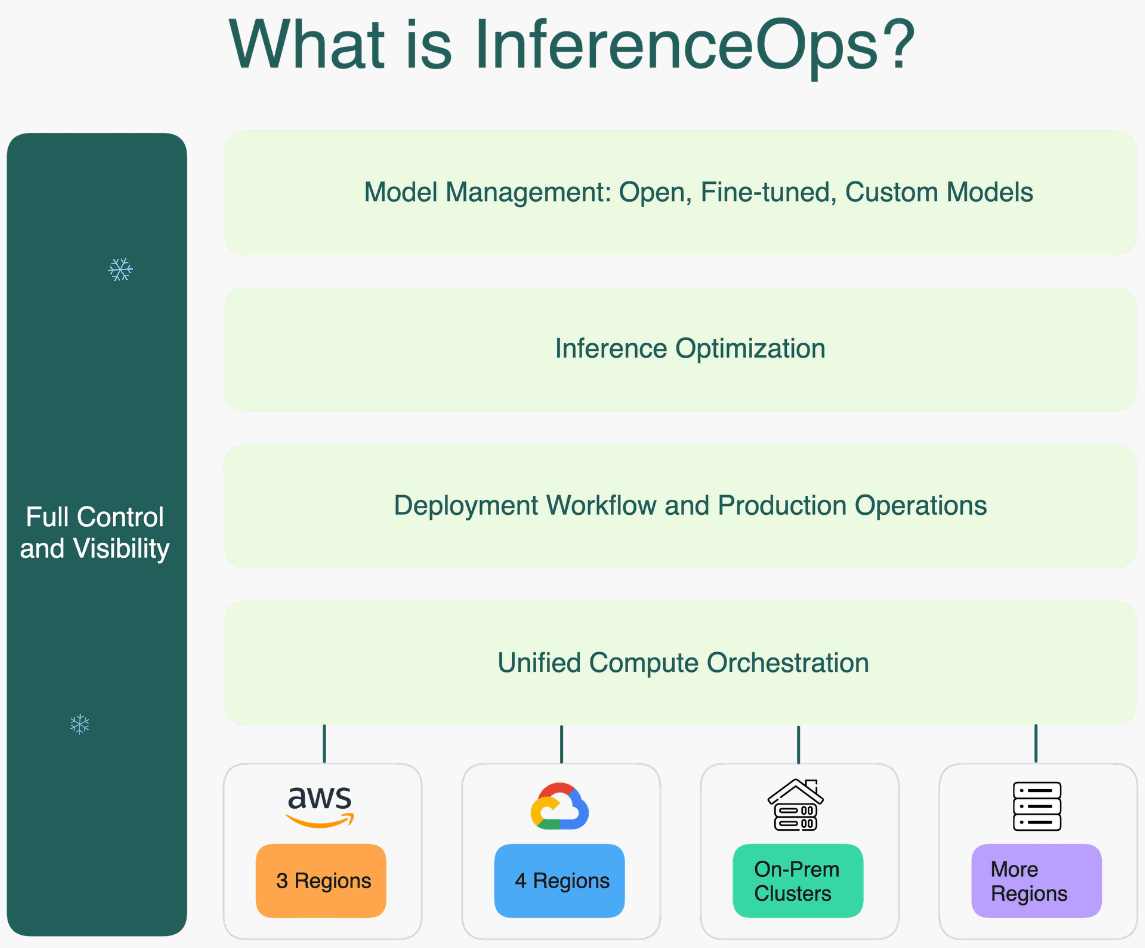
InferenceOps: A mindset shift

InferenceOps is a set of best practices and operational principles for running, scaling, and managing inference reliably and efficiently in production.

It represents a mindset shift: from treating inference as an afterthought to managing it as a critical layer of the modern AI stack. This includes but is not limited to:

- Continuously and rapidly iterating on models and inference code
- Optimizing inference performance and cost across varied workloads
- Ensuring reliability across heterogeneous models and distributed compute environments

This shift empowers AI teams to move faster and build differentiated AI systems to better serve their customers.


What is InferenceOps?
Full Control and Visibility
Model Management: Open, Fine-tuned, Custom Models
Inference Optimization
Deployment Workflow and Production Operations
Unified Compute Orchestration
aws
3 Regions
4 Regions
On-Prem Clusters
More Regions

Why InferenceOps matters now more than ever?

AI is no longer optional; it's eating the world. Every business that wants to compete today needs AI at its core. While choosing the right model is crucial, it's your inference layer that actually delivers that model's value to users. A strategy built solely on third-party LLM APIs can get you started quickly, but it won't sustain a differentiated, production-grade system.

Why relying on external LLM APIs falls short:

to your proprietary data and domain.

- **Zero control over the model:** You can't lock in a specific model version, roll back, or meet strict audit requirements.
- **Limited performance tuning:** You can't apply inference optimizations uniquely tailored to your SLAs or workload characteristics, align with your specific latency, throughput, and concurrency requirements.
- **Cost inefficiency at scale:** Shared-API pricing and opaque billing make it impossible to optimize per-request costs.

For a deeper breakdown, see Serverless vs. Dedicated LLM Deployments: A Cost-Benefit Analysis.

And that lack of customization hamstrings your ability to build lasting advantage:

1. **Compound AI systems** are where you win — chaining & composing models into richer workflows. See A Guide to Compound AI Systems for details.
2. **Tailored inference stacks** let you architect for precise SLAs and cost targets across different workloads.
3. **Fine-tuning and custom models** drive domain-specific accuracy and intellectual property capture.

Ultimately, **inference quality is product quality**. Mission-critical AI systems demand rock-solid performance, predictable costs, and airtight security, which only owning your inference infrastructure can deliver.

What are the core pillars of InferenceOps?

Like DevOps, InferenceOps isn't defined by a single tool or role; it's a set of operational principles. At its core, it shares many familiar foundations with DevOps, including:

- Automation and CI/CD pipelines
- System observability, monitoring and alerting
- Reliability with versioning, rollbacks, and safe deployments

These common practices are just the baseline. InferenceOps comes with unique requirements, especially for running and scaling LLMs in production.

Heterogeneous workflow management

In practice, enterprise AI is rarely powered by just one model. Different teams, products, and use cases often require different LLMs, each with distinct configurations and optimizations for optimal performance. One use case may need long-context summarization, and yet another may rely on high-throughput batch processing.

And it doesn't stop at LLMs. You'll often see small language models (SLMs), embedding models, fine-tuned models, and custom models running alongside LLMs in modern AI systems.

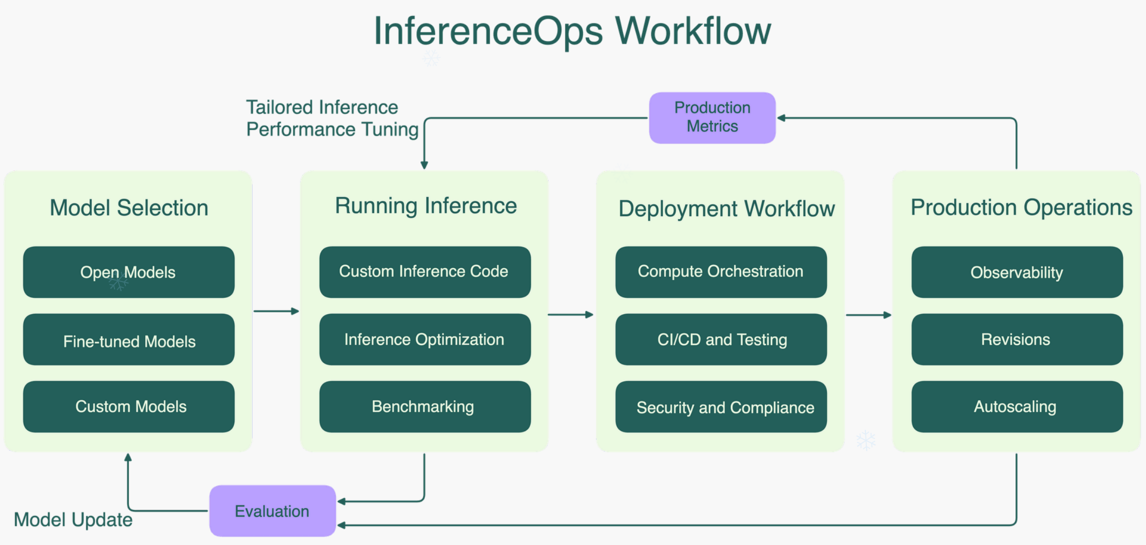
This results in a complex web of inference workflows around LLMs, which can quickly become operational sprawl.

A unified InferenceOps platform should provide a standardized way to manage diverse inference workflows. That means:

- Coordinating deployments across diverse model types

📢 Introducing llm-optimizer & LLM Performance Explorer — benchmark and optimize LLM inference with ease.

As your LLM inference scales, this kind of unification becomes more important. Without it, each new use case adds operational debt. With it, teams can scale responsibly, innovate faster, and maintain control.



Unified compute across any infrastructure

Different LLMs require different types of compute to deliver optimal performance. Some workloads are compute-bound, others memory-bound. Some benefit from high-throughput batching on A100s, while others demand low-latency responses on edge devices. Many AI teams even combine CPUs, GPUs, and I/O-heavy workloads in the same inference pipeline for complex AI systems.

To balance availability and cost, as well as to avoid vendor lock-in, AI teams often spread inference workloads across multiple cloud regions/providers and on-prem GPU vendors. This is especially true when dealing with GPU scarcity, region-specific latency requirements, or high GPU pricing to ensure performance and control cost.

But with flexibility comes complexity. A compute-agnostic InferenceOps platform should enable AI teams to:

- Run, scale, and manage inference seamlessly across hybrid compute environments
- Optimize GPU cost and dynamically match workloads to the most suitable hardware to maximize efficiency
- Avoid vendor lock-in while maintaining velocity and cost efficiency in acquiring and scaling compute

Inference workload optimization

Running LLM inference isn't just about launching a model server. It's an ongoing optimization effort in engineering for better performance and cost-efficiency.

The first challenge starts at build time. Many models require compilation or runtime optimization before serving, using tools like TensorRT-LLM and TorchScript. Each model architecture comes with its own quirks. Getting this right takes effort and skipping it leaves performance on the table.

Then there's scaling complexity. LLM workloads are not like typical web apps. They involve provisioning compute, pulling large container images, and loading model into GPUs, which makes cold starts slow and painful. See Scaling AI Models Like You Mean It for details.

- Prefill-decode disaggregation
- Prefix and KV cache aware routing
- KV cache management
- Tensor, pipeline, and expert parallelism techniques

They bring better resource allocation, smarter GPU usage, lower token latency, and reduced cost per token.

However, distributed LLM inference is complex by nature. Simply spinning up more Pods on Kubernetes won't cut it. You need purpose-built infrastructure for distributed LLM workloads. Done poorly, it leads to an over-provisioned, under-optimized system, wasting expensive GPUs while delivering sub-par performance.

A developer-friendly InferenceOps platform should abstract away the complexity and provide out-of-the-box support for these advanced optimizations.

Fast iteration without compromising reliability

From real-time chatbots to coding assistants, LLM inference now sits on the hot path of user-facing systems. That means production-grade reliability isn't optional. You need:

- Strict SLAs and uptime guarantees, especially for latency-sensitive workloads
- LLM-specific observability, going beyond generic system metrics to include:
  - Time to First Token (TTFT) and Time per Output Token (TPOT)
  - Tokens per Second (TPS) and Requests per Second (RPS)
  - Cost per request/token
  - Prompt filtering, output anomalies, model behavior drift
- Robust and automated pipelines, so your team can ship new models and code updates with zero downtime

But reliability often slows down velocity. Building this kind of infrastructure from scratch is slow, expensive, and error-prone. It requires specialized knowledge in GPU orchestration, observability stacks, and distributed inference systems. This means engineers have to spend more time fighting infrastructure than experimenting or shipping new ideas.

A comprehensive InferenceOps platform resolves this tension by providing:

- Native support for LLM-specific observability and monitoring
- Easy way to launch benchmark load testing to identify performance bottlenecks
- Optimized CI/CD for updates to models, adapters, and inference logic
- Built-in safeguards for deployments, rollbacks, and canary releases

With operational burdens offloaded, AI teams can move faster without compromising efficiency, reliability, and security.

Conclusion

📢 Introducing llm-optimizer & LLM Performance Explorer — benchmark and optimize LLM inference with ease.

- Run and scale inference reliably across heterogeneous compute environments
- Continuously optimize performance and cost with workload-specific techniques
- Accelerate iteration from development to production without sacrificing reliability

At Bento, we help enterprises run LLMs on their terms. With a distributed architecture and tailored inference optimization, our Bento Inference Platform enables you to deploy, scale, and manage any LLM from a single, unified interface.

- Talk to us about bringing InferenceOps into your LLM strategy
- Join our Slack community to connect with others building custom LLM applications
- Sign up to try the Bento Inference Platform and start running production-grade inference today
- Learn more about inference in our LLM Inference Handbook

Subscribe to our newsletter

Stay updated on AI infrastructure, inference techniques, and performance optimization.

Enter your email Subscribe

📢 Introducing llm-optimizer & LLM Performance Explorer — benchmark and optimize LLM inference with ease.

BentoML

LinkedIn

X

BentoML