ChatGPT

Hosting Infrastructure

For a managed or “Level 1” SaaS, mainstream cloud platforms are safest. AWS, GCP (and Azure) all offer GPU instances (e.g. EC2 P3/P4/P5 on AWS; A2 VMs or TPUs on GCP) with mature tools and global availability [1] [2]. Specialized GPU clouds (RunPod, Lambda Labs, CoreWeave, etc.) give per-second on-demand billing and the latest NVIDIA GPUs (A100, H100, etc.) often at lower cost [3] [1]. For example, Runpod supports on-demand A100/H100 pods with per-second pricing [3] and Lambda Labs offers on-demand A100/H100 servers with enterprise support [4]. In a Level 2 scenario, clients could choose any provider; our role is to supply Terraform/Helm templates so they can deploy on AWS, GCP, RunPod, Lambda, etc. Level 3 is fully custom integration.

Deployment Stack

Use containers, orchestration, and IaC to standardize deployments. Package each AI service (model server, API, etc.) in Docker images for reproducibility [5]. Manage infrastructure via Terraform: e.g. spin up GPU-enabled VMs or Kubernetes clusters on AWS/GCP and run containers there [6]. For orchestration, Kubernetes is a natural choice: define Deployments/DaemonSets for GPU workloads, use node pools with NVIDIA GPU device plugins, and automate via Helm or GitOps. (Tutorials show automating a K8s GPU cluster setup with Terraform and CI/CD tools [7].) Together, Docker+Kubernetes+Terraform give a portable, scalable stack.

Open-Source vs Commercial Tools

Favor open-source for core infra (Linux, Docker, Kubernetes, Terraform, Prometheus/Grafana for metrics, etc.). For custom logic or orchestration, use our existing OSS (e.g. the “switchyard” agentic framework). For **metering and billing**, it’s reasonable to use a third-party solution. Several open-source or SaaS tools target usage-based billing: for example, **OpenMeter** (MIT/Apache-licensed) provides usage metering, analytics, and Stripe integration out of the box [8]. **Lago** is an open-source billing API (AGPL) designed for usage-based and subscription models [9]. On the commercial side, products like **Amberflo** offer a hosted AI-finops platform with real-time token tracking and chargebacks [10], and **Metering.ai** automates pushing raw usage to Stripe for billing [11]. We can standardize around one choice for Level 2/3 (e.g. deploy OpenMeter via its Helm chart on the cluster) or simply integrate with Stripe’s built-in metered billing.

AI Usage Metering and Billing Integration

Because AI APIs are often metered (tokens, compute seconds, etc.), the solution should track usage per customer and convert it to bills. Stripe now natively supports metered billing: “Stripe’s usage-based billing lets you meter, ingest data, bill users, and process payments all in one place” [12]. If we use Stripe subscriptions, we can send usage records to Stripe directly. Alternatively, an external metering service (like Metering.ai) can aggregate logs or counters into Stripe-friendly usage events [11]. OpenMeter or Lago could serve as self-hosted middlewares: they ingest “events” or logs of API calls and compute billing metrics.

OpenMeter, for instance, advertises itself as flexible billing/metering for AI/DevTool companies with Stripe integration [8]. Lago is an open, event-based billing engine supporting hybrid pricing models [9]. These tools let you define units (e.g. tokens, calls) and pricing. In summary, plan to emit usage metrics (tokens, inference time, etc.) from each model endpoint into a metering system. For full billing integration, Stripe (or Chargebee/Recurly) can then invoice customers based on the aggregated usage.

Summary: Use cloud GPUs (AWS/GCP or AI-focused clouds) with Docker+Kubernetes+Terraform as a standardized stack. Leverage open-source OSS for orchestration, and choose an AI-specific metering tool (OpenMeter, Lago, Amberflo, or Stripe+Metering.ai) to record token usage and integrate with billing [8] [12]. This gives automated metered billing without reinventing the wheel.

Sources: Industry docs and tools for AI DevOps, including OpenMeter [8], Stripe Billing [12], cloud GPU comparisons [1] [3], and community guides on Docker/Terraform/K8s deployments [5] [7].

---

[1] [2] [3] [4] Top 12 Cloud GPU Providers for AI and Machine Learning in 2025
https://www.runpod.io/articles/guides/top-cloud-gpu-providers

[5] [6] From Beginner to Pro: Docker + Terraform for Scalable AI Agents - DEV Community
https://dev.to/docker/from-beginner-to-pro-deploying-scalable-ai-workloads-with-docker-terraform-41f2

[7] Kubernetes Meets Llama 3.2: How to Deploy AI Models on GPU Clusters - Civo.com
https://www.civo.com/learn/kubernetes-llama-deploy-ai-models-llm-gpu-cluster

[8] GitHub - openmeterio/openmeter: Metering and Billing for AI, API and DevOps. Collect and aggregate millions of usage events in real-time and enable usage-based billing.
https://github.com/openmeterio/openmeter

[9] GitHub - getlago/lago: Open Source Metering and Usage Based Billing API ⭐ Consumption tracking, Subscription management, Pricing iterations, Payment orchestration & Revenue analytics
https://github.com/getlago/lago

[10] Amberflo | FinOps for AI
https://www.amberflo.io/

[11] Upload Usage Data, Connect to Stripe, Complete Billing | Metering.ai
https://www.metering.ai/

[12] stripe.com
https://stripe.com/llms.txt