AI Conexio

DEPLOYMENT GUIDE
2026 Edition

SELF-HOSTED AI INFRASTRUCTURE

Private LLM Deployment Guide

Deploy Large Language Models on your own infrastructure without sharing data with third parties. Complete guide to vLLM, infrastructure planning, security, and production deployment.

Prepared by AI Conexio Engineering & Advisory

Schedule a Strategy Call Now →

What This Guide Covers

A complete, end-to-end playbook for taking an open-source LLM from evaluation to production on infrastructure you control. Each section builds on the previous one, from choosing a serving platform through hardening, tuning, and operating the deployment.

Introduction: The Case for Private AI 3

1. Understanding Your Deployment Options 4

2. Architecture & Infrastructure Planning 5

3. Implementation Roadmap 6–7

4. Security & Compliance 8–9

5. Model Selection & Fine-Tuning 10–11

6. Cost Analysis & ROI 12

7. Performance Optimization 13

8. Production Best Practices 14

Frequently Asked Questions 15

Next Steps 16

WHO THIS GUIDE IS FOR

Technical leaders, platform engineers, and DevOps teams evaluating or building self-hosted LLM infrastructure. It assumes general familiarity with cloud or data center operations, but no prior experience serving LLMs. Organizations processing sensitive data, requiring HIPAA/GDPR compliance, or operating at high volume (1M+ requests/month) will find the cost and security sections directly actionable.

© 2026 AI Conexio

Introduction: The Case for Private AI

While cloud-based LLM APIs offer convenience, many enterprises require complete control over their AI infrastructure. Private LLM deployment ensures **data never leaves your infrastructure**, enables **unlimited customization**, and can be **more cost-effective at scale**.

WHY DEPLOY PRIVATE LLMS?

Organizations processing sensitive data, requiring HIPAA/GDPR compliance, or operating at high volume (1M+ requests/month) often find private deployment necessary and cost-effective.

KEY BENEFITS OF PRIVATE DEPLOYMENT

Complete Data Privacy
Sensitive data never leaves your infrastructure

Regulatory Compliance
Meet HIPAA, GDPR, SOC 2 requirements

Cost Predictability
Fixed infrastructure costs vs. per-token pricing

Performance Control
Optimize latency and throughput for your needs

Model Customization
Fine-tune and modify models without restrictions

No Vendor Lock-in
Switch between open-source models freely

Unlimited Requests
No rate limits or throttling

Offline Capability
Operate without internet connectivity

IMPORTANT CONSIDERATIONS

Private deployment requires technical expertise, upfront infrastructure investment, and ongoing maintenance. Evaluate whether the benefits justify the complexity for your use case. For many organizations, a hybrid approach (API for development, private for production) offers the best balance.

© 2026 AI Conexio

1. Understanding Your Deployment Options

Multiple solutions exist for private LLM deployment, each optimized for different use cases and technical requirements. Choose based on your scale, expertise, and infrastructure.

INFERENCE PLATFORMS COMPARISON

PLATFORM	ADVANTAGES	LIMITATIONS	IDEAL USE CASE
vLLM High-performance inference server with advanced batching	Fastest inference speed; PagedAttention for efficiency; continuous batching; OpenAI-compatible API	Steeper learning curve; requires GPU infrastructure	**Best for:** production deployments requiring maximum throughput. 1M+ requests/month, low-latency requirements.
Ollama Developer-friendly local LLM runtime with simple interface	Easiest to get started; great developer experience; model library included; CPU support	Limited scalability; less production-focused	**Best for:** development, prototyping, and smaller deployments. Development environments, single-user applications.
Text Generation Inference (TGI) Hugging Face's production-ready inference toolkit	Easy Hugging Face integration; good documentation; built-in monitoring; quantization support	Not as fast as vLLM; opinionated architecture	**Best for:** balanced ease-of-use and performance. 100K to 1M requests/month, Hugging Face ecosystem.
TensorRT-LLM NVIDIA's optimized inference engine for maximum performance	Best GPU utilization; lowest latency; advanced optimizations; multi-GPU support	NVIDIA GPUs only; complex setup; limited model support	**Best for:** highest performance needs with NVIDIA GPUs. Ultra-low latency, NVIDIA infrastructure.

INFRASTRUCTURE DEPLOYMENT MODELS

MODEL	DESCRIPTION	ADVANTAGES	TRADEOFFS
On-Premise	Own hardware in your data center	Maximum control; no cloud costs; lowest latency	High upfront cost; maintenance burden; scaling difficulty
Private Cloud	AWS VPC, Azure Private, GCP	Scalable; managed infrastructure; quick setup	Ongoing costs; some vendor dependency; compliance complexity
Hybrid	Combination of on-prem and cloud	Flexibility; cost optimization; disaster recovery	Complex management; network overhead; higher expertise needed

© 2026 AI Conexio

2. Architecture & Infrastructure Planning

Proper infrastructure planning ensures optimal performance and cost-efficiency. Hardware requirements vary dramatically based on model size and expected throughput.

GPU REQUIREMENTS BY MODEL SIZE

MODEL SIZE	EXAMPLE MODELS	MIN VRAM (FP16)	MIN VRAM (4-BIT)	RECOMMENDED GPU
7B parameters	Llama 3 8B, Mistral 7B	16 GB	6 GB	RTX 4090, L40S
13B parameters	Llama 2 13B, Vicuna 13B	28 GB	10 GB	A40, A100 40GB
34B parameters	Yi 34B, Mixtral 8x7B	70 GB	24 GB	A100 80GB, 2× A40
70B parameters	Llama 3 70B, Falcon 70B	140 GB	45 GB	2× A100 80GB, H100
405B parameters	Llama 3.1 405B	810 GB	200 GB	8× H100 80GB

MEMORY CALCULATION FORMULA

```
VRAM Required = Model Parameters × Precision (bytes) × 1.2 (overhead)
```

- ✓ **FP16:** 2 bytes per parameter (70B model = 140 GB + 20% overhead = 168 GB)
- ✓ **INT8:** 1 byte per parameter (70B model = 70 GB + 20% = 84 GB)
- ✓ **4-bit (GPTQ/AWQ):** 0.5 bytes per parameter (70B model = 35 GB + 20% = 42 GB)

COMPLETE INFRASTRUCTURE STACK

COMPUTE LAYER

- ✓ GPU: NVIDIA A100, H100, L40S, or RTX 4090 (for smaller models)
- ✓ CPU: 32+ cores for handling concurrent requests
- ✓ RAM: 256 GB+ system memory for large batch processing
- ✓ Cooling: enterprise-grade for sustained high utilization

STORAGE LAYER

- ✓ SSD: 1TB+ NVMe for model storage and fast loading
- ✓ Model cache: 500 GB per major model variant
- ✓ Logs & metrics: 100 GB per month
- ✓ Backup: 2-3× model storage for redundancy

NETWORK LAYER

- ✓ Bandwidth: 10 Gbps+ for multi-GPU communication
- ✓ Low latency: under 1ms between GPU nodes
- ✓ Load balancer: NGINX or HAProxy for distribution
- ✓ VPN: site-to-site for hybrid deployments

MONITORING & OBSERVABILITY

- ✓ Metrics: Prometheus + Grafana for GPU utilization
- ✓ Logging: ELK or Loki for centralized logs
- ✓ Tracing: Jaeger for request tracking
- ✓ Alerting: PagerDuty or Opsgenie integration

© 2026 AI Conexio

3. Implementation Roadmap

Follow a phased approach to minimize risk and demonstrate value early. Each phase builds on the previous one with increasing scale and sophistication.

PHASE 1: PROOF OF CONCEPT (WEEKS 1-4)

Goal: validate technical feasibility and basic performance.

KEY TASKS

- ✓ Set up single-GPU development environment
- ✓ Deploy Ollama or vLLM with 7B model
- ✓ Build basic API integration
- ✓ Benchmark inference speed and accuracy
- ✓ Test with real use cases (100-1,000 requests)

DELIVERABLES

- ✓ Working prototype on single GPU
- ✓ Performance benchmark report
- ✓ Cost projection for production scale
- ✓ Technical architecture document

Success criteria: sub-second response time, 90%+ accuracy vs. baseline.

PHASE 2: PRODUCTION DEPLOYMENT (WEEKS 5-12)

Goal: deploy scalable production infrastructure.

KEY TASKS

- ✓ Set up multi-GPU production cluster
- ✓ Implement load balancing and redundancy
- ✓ Configure monitoring and alerting
- ✓ Establish security and access controls
- ✓ Deploy to staging for testing
- ✓ Conduct load testing (100K+ requests)
- ✓ Create runbooks and documentation

DELIVERABLES

- ✓ Production-ready cluster
- ✓ Monitoring dashboards
- ✓ Security audit report
- ✓ Operational runbooks

Success criteria: 99.9% uptime, handles peak load, all security controls active.

© 2026 AI Conexio

Implementation Roadmap (continued)

PHASE 3: OPTIMIZATION & FINE-TUNING (WEEKS 13-20)

Goal: maximize performance and reduce costs.

KEY TASKS

- ✓ Fine-tune model on domain-specific data
- ✓ Implement quantization (4-bit) for efficiency
- ✓ Optimize batching and caching strategies
- ✓ Set up auto-scaling based on load
- ✓ Conduct A/B testing vs. baseline
- ✓ Measure and improve GPU utilization

DELIVERABLES

- ✓ Fine-tuned model with improved accuracy
- ✓ Optimized infrastructure (30-50% cost reduction)
- ✓ Auto-scaling policies
- ✓ Performance optimization report

Success criteria: 2x throughput, 40% cost reduction, improved accuracy.

PHASE 4: SCALE & EXPAND (ONGOING)

Goal: scale to full production and new use cases.

KEY TASKS

- ✓ Roll out to all users and applications
- ✓ Add additional models for different use cases
- ✓ Establish model versioning and deployment pipeline
- ✓ Build internal expertise and training
- ✓ Continuous monitoring and improvement
- ✓ Expand to new geographic regions if needed

DELIVERABLES

- ✓ Multi-model production deployment
- ✓ CI/CD pipeline for model updates
- ✓ Training program for internal teams
- ✓ Disaster recovery plan

Success criteria: supports 1M+ monthly requests, multiple use cases live.

© 2026 AI Conexio

4. Security & Compliance

Private LLM deployment provides the foundation for strong security, but proper implementation is essential. Address network isolation, access controls, compliance requirements, and audit trails.

SECURITY ARCHITECTURE LAYERS

NETWORK SECURITY

Network Isolation: deploy in private VLAN/VPC with no internet access

VPN Access: require VPN for all connections to LLM endpoints

Firewall Rules: whitelist only necessary ports and IPs

TLS Encryption: enforce TLS 1.3 for all communications

ACCESS CONTROL

Authentication: OAuth 2.0 or SAML for user authentication

Authorization: role-based access control (RBAC) with least privilege

API Keys: rotate keys every 90 days, scope to specific services

MFA: require multi-factor auth for admin access

DATA PROTECTION

Encryption at Rest: AES-256 encryption for model storage and logs

Encryption in Transit: TLS 1.3 for all data transmission

Input Sanitization: validate and sanitize all inputs to prevent injection

Output Filtering: filter PII and sensitive data from responses

AUDIT & COMPLIANCE

Audit Logging: log all requests with user ID, timestamp, and action

Log Retention: retain logs for 1-7 years based on compliance needs

Compliance Scanning: regular vulnerability scans and penetration testing

Incident Response: documented playbooks for security incidents

© 2026 AI Conexio

Compliance Framework Checklist

Map your deployment controls to the regulatory frameworks that apply to your organization. Use each checklist as the baseline scope for your audit preparation.

HIPAA COMPLIANCE

- [ ] Encrypt all PHI data at rest and in transit
- [ ] Maintain audit logs for all data access
- [ ] Implement Business Associate Agreements (BAA)
- [ ] Conduct annual risk assessments
- [ ] Train staff on HIPAA requirements
- [ ] Establish breach notification procedures

GDPR COMPLIANCE

- [ ] Implement data minimization principles
- [ ] Enable right to erasure (delete user data)
- [ ] Conduct Data Protection Impact Assessments
- [ ] Appoint Data Protection Officer if required
- [ ] Document data processing activities
- [ ] Ensure data portability capabilities

SOC 2 TYPE II

- [ ] Establish and document security policies
- [ ] Implement change management procedures
- [ ] Monitor system availability (99.9%+ uptime)
- [ ] Conduct regular security audits
- [ ] Maintain incident response plan
- [ ] Implement continuous monitoring

ISO 27001

- [ ] Establish Information Security Management System
- [ ] Conduct risk assessment and treatment
- [ ] Implement security controls framework
- [ ] Regular internal audits
- [ ] Management review processes
- [ ] Continuous improvement procedures

KEEPING DATA INSIDE YOUR PERIMETER

Deploy in a private VPC/VLAN with no internet access, require VPN for all connections, and disable model telemetry and logging to external services. Audit network traffic regularly, and use air-gapped deployment for maximum security. All inference happens locally; no data is sent to external APIs. Document your data flow in compliance reports.

© 2026 AI Conexio

5. Model Selection & Fine-Tuning

Choose the right open-source model based on your use case, then optimize through quantization and fine-tuning. Balance model capability, inference speed, and resource requirements.

POPULAR OPEN-SOURCE MODELS (2025)

MODEL	SIZE	LICENSE	BEST FOR	PERFORMANCE
Llama 3.1	8B, 70B, 405B	Meta License	General purpose, instruction following	**Excellent**
Mistral 7B	7B	Apache 2.0	Fast inference, efficient	**Excellent**
Mixtral 8x7B	47B (MoE)	Apache 2.0	High quality, cost-effective	**Excellent**
Qwen 2.5	7B, 14B, 72B	Apache 2.0	Multilingual, long context	**Excellent**
DeepSeek Coder	6.7B, 33B	MIT License	Code generation & analysis	**Very Good**
Falcon 180B	180B	Apache 2.0	Maximum capability	**Excellent**

QUANTIZATION STRATEGIES

GPTQ

4-bit quantization with minimal accuracy loss

Memory: 75% reduction
Accuracy: 1-2% loss
Speed: fast inference
Best for: production deployments

AWQ

Activation-aware weight quantization

Memory: 75% reduction
Accuracy: under 1% loss
Speed: faster than GPTQ
Best for: high-accuracy needs

GGUF

CPU-friendly quantization format

Memory: 50-75% reduction
Accuracy: varies by precision
Speed: good CPU performance
Best for: CPU deployment

Quantization is the highest-leverage cost lever in private deployment: a 70B model that needs 168 GB of VRAM at FP16 fits in roughly 42 GB at 4-bit, cutting the GPU bill by up to 75% for a typically negligible accuracy tradeoff.

© 2026 AI Conexio

Fine-Tuning Approaches

Four approaches span the cost and customization spectrum. Most production teams land on LoRA; reserve full fine-tuning for unique domains with large datasets.

FULL FINE-TUNING · High ($10K-$50K+) · 2-4 weeks · 10K-100K examples

Update all model parameters on your data.

Advantages: maximum customization; best accuracy; full control

Limitations: expensive; requires expertise; long training time

Best use case: unique domain with large dataset

LORA (LOW-RANK ADAPTATION) · Medium ($1K-$10K) · 3-7 days · 1K-10K examples

Train small adapter layers, keep base model frozen.

Advantages: cost-effective; fast training; multiple adapters

Limitations: slightly lower accuracy; limited customization

Best use case: most production use cases

QLORA · Low ($500-$5K) · 2-5 days · 1K-10K examples

LoRA with quantization for even lower memory.

Advantages: very cost-effective; single GPU training; fast iteration

Limitations: some accuracy tradeoff; quantization overhead

Best use case: budget-conscious deployments

FEW-SHOT PROMPTING · Very Low (under $500) · 1-3 days · 10-100 examples

No training, just engineered prompts with examples.

Advantages: no training needed; instant results; easy updates

Limitations: lower accuracy; limited customization; prompt engineering required

Best use case: quick prototypes, simple tasks

© 2026 AI Conexio

6. Cost Analysis & ROI

Private LLM deployment involves higher upfront costs but becomes more economical at scale. Calculate your break-even point based on request volume and infrastructure choices.

INFRASTRUCTURE COST BREAKDOWN

COMPONENT	OPTION	INITIAL COST	MONTHLY COST	NOTES
GPU Server	1× A100 40GB	$15K	$1,500/mo	Cloud or $8K purchase + hosting
	2× A100 80GB	$30K	$3,000/mo	For 70B models
	8× H100 80GB	$200K	$15,000/mo	For 405B models
Storage	2TB NVMe SSD	$1K	$100/mo	Model storage + cache
Networking	Load balancer + VPN	$500	$200/mo	Traffic costs extra
Monitoring	Grafana + Prometheus	$0	$0-$500/mo	Self-hosted or managed
Personnel	1 FTE DevOps	$0	$8K-$15K/mo	Maintenance & optimization
Total (Medium Deployment)		**$31.5K**	**$5K-$12K/mo**	**2× A100 80GB setup**

BREAK-EVEN ANALYSIS: PRIVATE VS. API

SCENARIO	VOLUME	API COST	PRIVATE COST	BREAK-EVEN	RECOMMENDATION
Customer Support Bot	100K requests/month	$1,500/mo	$5,000/mo initial	Month 1 (if >350K req/mo)	Use API initially, switch at 500K+ req/mo
Document Processing	1M requests/month	$15,000/mo	$8,000/mo	Month 5	Private deployment recommended
Internal Chatbot	10K daily users (300K/mo)	$4,500/mo	$6,000/mo	Month 3 (if usage grows)	Hybrid: API dev, private production

COST OPTIMIZATION TIPS

- ✓ Use 4-bit quantization to reduce GPU requirements by 75%
- ✓ Implement aggressive caching to reduce redundant inference
- ✓ Auto-scale GPUs based on demand (spin down during off-hours)
- ✓ Use spot instances or reserved instances for 40-70% cloud savings
- ✓ Choose smaller specialized models over large general models when possible

© 2026 AI Conexio

7. Performance Optimization

Maximize throughput and minimize latency through advanced optimization techniques. Proper optimization can double throughput while reducing costs by 40-50%.

KEY OPTIMIZATION STRATEGIES

STRATEGY & IMPACT	IMPLEMENTATION STEPS	SUCCESS METRICS
Continuous Batching **2-5x throughput increase** Process multiple requests in parallel by batching at the token level	1. Enable in vLLM with the `--max-batch-size` parameter 2. Start with batch size 32-64, tune based on GPU memory 3. Monitor queue depth and adjust dynamically 4. Use priority queues for time-sensitive requests	Target: >80% GPU utilization, <2s p95 latency
KV Cache Optimization **30-50% memory savings** Efficiently manage key-value cache to handle longer contexts	1. Use PagedAttention (built into vLLM) for efficient memory management 2. Set appropriate KV cache size based on max context length 3. Enable prefix caching for repeated prompts 4. Implement cache eviction policies for long-running instances	Target: support 4K+ context at 90%+ GPU memory utilization
Model Parallelism **Enable larger models, reduce latency** Split model across multiple GPUs for large models	1. Use tensor parallelism for large models (70B+) 2. Pipeline parallelism for multi-node deployments 3. Optimize communication with NCCL/RCCL 4. Balance load across GPUs evenly	Target: linear scaling with GPU count (e.g., 2× GPUs = 1.8× throughput)
Speculative Decoding **1.5-2x faster generation** Use smaller draft model to predict tokens, verify with main model	1. Deploy small draft model (7B) alongside main model (70B) 2. Draft model generates 3-5 token candidates 3. Main model verifies in parallel 4. Fallback to standard decoding if draft fails	Target: 50%+ faster generation for long outputs
Caching & Memoization **60-80% cost reduction for repeated queries** Cache responses for identical or similar inputs	1. Implement semantic caching with embedding similarity 2. Cache at multiple levels (exact match, fuzzy match, prefix) 3. Use Redis or Memcached for distributed caching 4. Set TTL based on content freshness requirements	Target: >40% cache hit rate, <10ms cache lookup

© 2026 AI Conexio

8. Production Best Practices

Running private LLMs in production requires robust monitoring, incident response, and continuous improvement processes. Follow these practices to ensure reliability and performance.

MONITORING & ALERTING

- ✓ Track GPU utilization, memory, temperature continuously
- ✓ Monitor request latency (p50, p95, p99)
- ✓ Alert on queue depth over 100 requests
- ✓ Track model accuracy drift over time
- ✓ Monitor error rates and types
- ✓ Set up uptime monitoring with PagerDuty

DEPLOYMENT & UPDATES

- ✓ Use blue-green deployment for zero-downtime updates
- ✓ Test model changes in staging environment first
- ✓ Implement gradual rollout (10% → 50% → 100%)
- ✓ Keep rollback plan ready (previous model warm)
- ✓ Version all models with semantic versioning
- ✓ Document all changes in changelog

INCIDENT RESPONSE

- ✓ Create runbooks for common issues (OOM, slow inference)
- ✓ Set up automated failover to backup infrastructure
- ✓ Implement circuit breakers for cascading failures
- ✓ Conduct post-mortems for all major incidents
- ✓ Practice disaster recovery quarterly
- ✓ Maintain on-call rotation with clear escalation

CAPACITY PLANNING

- ✓ Track growth trends and forecast capacity needs
- ✓ Maintain 30% headroom for traffic spikes
- ✓ Plan for 3-6 month infrastructure lead times
- ✓ Load test at 2x expected peak traffic
- ✓ Document resource limits and scaling procedures
- ✓ Review capacity monthly with stakeholders

THE OPERATING RHYTHM

Treat the LLM platform like any other tier-1 production system: dashboards reviewed daily, capacity reviewed monthly, disaster recovery practiced quarterly, and every model change shipped through staging with a warm rollback path. Teams that adopt this rhythm in Phase 2 avoid nearly all of the incidents that derail private deployments in their first year.

© 2026 AI Conexio

Frequently Asked Questions

Common questions about private LLM deployment.

How much does it cost to run a private LLM?

Costs vary widely based on model size and scale. A small deployment (7B model, single GPU) costs $1,500-$3,000/month. Medium deployments (70B model, 2× A100) cost $8,000-$12,000/month including personnel. Large deployments (405B model, 8× H100) cost $20,000-$30,000/month. Break-even vs. API pricing typically occurs around 500K-1M requests/month.

Can we run LLMs without expensive GPUs?

Yes, but with significant performance tradeoffs. Quantized 7B models can run on CPU using Ollama or llama.cpp, but expect 10-100x slower inference. For production use cases requiring sub-second latency, GPUs are essential. Consider starting with cloud GPU rentals ($1.50-$3/hour) before purchasing hardware.

How do we ensure data never leaves our infrastructure?

Deploy in a private VPC/VLAN with no internet access. Use VPN for all connections. Disable model telemetry and logging to external services. Audit network traffic regularly. Use air-gapped deployment for maximum security. All inference happens locally; no data is sent to external APIs. Document your data flow in compliance reports.

What's the difference between vLLM and Ollama?

vLLM is optimized for high-throughput production deployments with advanced batching, while Ollama prioritizes ease of use and developer experience. vLLM offers 2-5x better throughput and supports OpenAI-compatible APIs, but requires more expertise to set up. Ollama is perfect for development and prototyping. Most organizations use Ollama for development and vLLM for production.

How long does implementation take?

Timeline depends on complexity and team expertise. Proof of concept: 2-4 weeks. Production deployment: 8-12 weeks. Full optimization: 16-20 weeks. Accelerate by partnering with experts who have done it before. Most organizations see initial results in 4-6 weeks with a phased approach.

Can we use open-source models commercially?

Most open-source LLMs (Llama 3, Mistral, Mixtral) allow commercial use, but check each model's license. Meta's Llama license permits commercial use. Apache 2.0 licensed models (Mistral, Falcon) have no restrictions. Some models restrict commercial use for organizations over certain revenue thresholds. Always review the license file in the model repository before deployment.

© 2026 AI Conexio

Ready to Take Your AI Stack Private?

You now have a complete framework for deploying private LLM infrastructure: platform selection, hardware sizing, security hardening, fine-tuning, and production operations. The fastest path to production is a phased rollout led by a team that has done it before.

HOW AI CONEXIO HELPS:

- ✓ Size and architect your GPU infrastructure (on-premise, private cloud, or hybrid)
- ✓ Stand up vLLM or TGI serving with load balancing, monitoring, and auto-scaling
- ✓ Harden your deployment for HIPAA, GDPR, SOC 2, and ISO 27001 requirements
- ✓ Fine-tune and quantize models to cut inference costs by up to 75%

Book a Strategy Call

Speak directly with an AI Infrastructure Architect to discuss your private deployment roadmap.

Schedule a Strategy Call Now →