BUILDING AI AGENTS WITH OLLAMA

DESIGN, DEPLOY, AND SCALE RELIABLE LOCAL AI SYSTEMS

COMPLETE CODE EXAMPLES

PRODUCTION READY PATTERNS

RUNS ENTIRELY ON YOUR HARDWARE

NO CLOUD REQUIRED

MICHAEL CLINTON

Building AI Agents with Ollama

Design, Deploy, and Scale Reliable Local AI Systems

Steve Publications

This book is available at https://leanpub.com/buildingaiagentswithollama

This version was published on 2026-07-29

This is a Leanpub book. Leanpub empowers authors and publishers with the Lean Publishing process. Lean Publishing is the act of publishing an in-progress ebook using lightweight tools and many iterations to get reader feedback, pivot until you have the right book and build traction once you do.

Contents

Design, Deploy, and Scale Reliable Local AI Systems

This book teaches you how to design, build, deploy, and maintain production-grade AI agents that run entirely on your own hardware using Ollama as the inference engine. You will learn everything from installation and model selection through advanced multi-agent architectures, RAG pipelines, security hardening, and observability. Every chapter includes complete, working code examples that you can copy, build, and run without modification. Whether you are building personal assistants, coding agents, document analysis tools, or autonomous workflow systems, this book provides the patterns, implementations, and engineering practices needed to make them work reliably in the real world.

Introduction: The Case for Local Agents

Why Local Matters Now

There was a moment in early 2024 when every serious developer asked the same question: where should my AI run? The cloud had obvious advantages. You could call GPT-4 or Claude with a few lines of code and get world-class intelligence without buying hardware, installing drivers, or understanding how neural networks worked. But that convenience came with costs that became impossible to ignore as AI moved from experimentation to production.

Consider a startup building an AI-powered document review tool: three enterprise clients each processing thousands of pages monthly can easily push token bills past $15,000 in a single month. Consider a financial services firm evaluating AI for client document analysis: sending customer financial records to any cloud API may violate GDPR, HIPAA, or industry-specific compliance rules. Consider a developer building a coding assistant that analyzes proprietary source code: every line sent to an external API is data you no longer fully control. These are not hypothetical edge cases. They are the natural consequences of building AI systems on infrastructure you do not control.

The economics tell a clear story. At current pricing (as of mid-2026), mainstream models like GPT-4o charge $2.50 per million input tokens and $10 per million output tokens, while premium reasoning models and earlier Claude Opus versions have charged up to $75 per million output tokens [1,7,16]. For agentic workflows that require dozens of API calls per task, costs scale linearly with complexity. Consider a document review tool processing thousands of pages for enterprise clients: each page analyzed might trigger multiple API calls for extraction, analysis, and summarization. Multiply that by daily volume across several clients, and monthly token bills easily exceed $10,000 to $20,000. Local inference changes this equation entirely. After a one-time hardware investment, your marginal cost per token is effectively zero. For consistent, heavy usage, the break-even point typically arrives within six to eight months, after which local AI becomes strictly cheaper than any cloud alternative [1,7,16].

But economics is only part of the argument. Privacy and data control are the stronger drivers for many organizations. Consider a financial services firm evaluating AI for client document analysis: sending customer financial records to any cloud API may violate regulatory requirements like GDPR, HIPAA, or industry-specific compliance rules. Or imagine a software company building a coding assistant that analyzes proprietary source code: every line sent to an external API potentially trains future competitors' models. When your AI runs locally, sensitive material never leaves your network. Customer records, internal documents, proprietary code, and personal communications stay on your machines. This is not a marketing claim but a structural property of the architecture. There is no telemetry, no logging to external servers, no risk that your data will be used to train a model you do not own. For regulated industries like healthcare, finance, and government, this is often the difference between being able to deploy AI and not [5,13].

Latency is another factor that becomes significant at scale. Cloud APIs introduce network round-trips for every request, typically adding 100 to 300 milliseconds of latency per call. For agentic workflows that may make dozens of API calls in sequence, this compounds into noticeable delays. Local inference on a modern GPU can generate tokens in real time with latencies measured in single-digit milliseconds, creating a fundamentally better user experience for interactive applications [7,16].

Perhaps most importantly, local AI gives you control over the stack. You choose the models, you update them when you want, and you are not at the mercy of API deprecations, pricing changes, or service outages. When OpenAI changed its terms of service in 2023, thousands of applications had to scramble to comply. When a cloud provider experiences an outage, your application stops working. With local inference, you own the failure modes, and they are ones you can engineer against [9,14].

The technology has matured to make this practical. Open-source models released in 2024 and 2025 have closed much of the performance gap with proprietary alternatives on many tasks. Meta Llama 3.1, Mistral, Qwen, and Google Gemma series all offer strong capabilities at sizes that fit on consumer hardware. Ollama, the inference engine this book focuses on, has emerged as the most popular way to run these models locally, with millions of downloads and a rapidly growing ecosystem [1,4].

This book is about building systems that work, not just prototypes that impress. Local AI introduces engineering challenges that cloud APIs hide from

you. You must manage hardware resources, handle model loading and unloading, optimize inference performance, design fault-tolerant architectures, and ensure security without the safety net of a managed service provider. The goal is to give you the complete toolkit to meet these challenges head on.

What Is an AI Agent

The term "agent" has been stretched to mean almost anything that uses an LLM, but there is a useful distinction worth making. A chatbot takes a prompt and returns a response. An agent takes a goal and pursues it, potentially using tools, maintaining state across multiple interactions, and adapting its strategy based on feedback from the environment [2,6].

Consider the difference between asking a model to summarize a document and asking it to analyze all PDFs in a folder, extract key metrics, generate a report, and email it to your team. The first is a single prompt-response cycle. The second requires the system to perceive its environment (list files), plan actions (which files to read, what to extract), execute operations (read files, call an email API), handle failures (what if a file is corrupted?), and maintain coherence across the entire workflow. That is agentic behavior.

The core capabilities that distinguish agents from simple chat interfaces are:

- **Goal-directed behavior**: The system works toward completing a task, not just responding to input.
- **Tool use**: The system can call external functions, APIs, databases, or other services to gather information or take action.
- **State and memory**: The system maintains context across multiple steps and interactions, remembering what it has done and learned.
- **Planning and reasoning**: The system decomposes complex tasks into subtasks, reasons about the best approach, and adapts when things go wrong.
- **Autonomy**: The system can operate without constant human guidance, making decisions within defined boundaries.

These capabilities are not magical. They are engineering choices implemented through specific architectural patterns. Tool calling is just structured output that triggers function execution. Memory is just persistent storage

queried at the right time. Planning is just prompting strategies that elicit step-by-step reasoning. What makes building agents hard is not any single capability but the interactions between them: how memory affects planning, how tool failures propagate, how to ensure the system stays on task without going off the rails.

Local agents add another layer of complexity because you are responsible for the entire stack. When you use a cloud provider's agent framework, they handle model scaling, load balancing, and basic reliability. Running locally, these become your problems to solve. This book treats that responsibility as an opportunity rather than a burden, because owning the stack means you can optimize it for your specific needs in ways no managed service can match [1,3,4].

The Reliability Gap

Here is the uncomfortable truth about most AI agent implementations: they are fragile. They work in demos and break in production. They handle happy paths gracefully but fall apart when faced with real-world edge cases like network timeouts, malformed tool responses, or ambiguous user requests. This fragility is not a flaw in the underlying models; it is a failure to treat agents as software systems that require the same engineering rigor as any other production application.

The reliability gap exists because AI development has historically borrowed more from research than from engineering. Researchers optimize for novelty and benchmark scores. Engineers optimize for correctness, maintainability, and user trust. Building reliable agents requires bridging this gap by applying proven software engineering practices to a domain that does not yet have established conventions.

Consider what can go wrong in even a simple agent:

- The model returns malformed JSON when expected to produce structured output.
- A tool call times out, leaving the agent in an inconsistent state.
- The context window fills up mid-conversation, causing the agent to forget critical information.
- A prompt injection attack tricks the agent into executing unauthorized actions.

- Concurrent requests overload the model, causing cascading failures.
- An update to a tool API silently changes its response format, breaking the agent.

Each of these is a known failure mode with established mitigation strategies. The problem is that most tutorials and documentation do not cover them. They show you how to make the agent work, not how to make it work reliably under stress, over time, and in the presence of adversarial inputs.

This book addresses the reliability gap directly. Every chapter emphasizes production-ready patterns: error handling that recovers gracefully, testing strategies that catch regressions, observability that lets you debug failures, and security practices that protect against attacks. The code examples are not minimal demos; they are implementations designed to survive contact with the real world.

The thesis running through this book is that local AI agents are a distinct engineering discipline, different from both traditional software development and cloud-based AI application development. They require new patterns for reliability, observability, and resource management that do not appear in conventional engineering textbooks or cloud provider documentation. The goal is to establish those patterns clearly, with working implementations you can adapt to your own systems [1,2,6,8].

How This Book Is Structured

This book progresses from foundation to mastery, with each chapter building on the previous ones. You do not need to read it linearly if you are already comfortable with some topics, but the chapters are organized to minimize gaps in understanding.

The **Foundations** section (Chapters 1-2) covers the technical prerequisites: how local LLM inference works, how to install and configure Ollama for production use, and how to select and manage models based on your task requirements and hardware constraints. If you are new to local AI, start here.

The **Core Capabilities** section (Chapters 3-6) covers the building blocks of agent behavior: prompt engineering patterns that produce reliable outputs, structured data extraction and tool calling, memory systems for maintaining

context, and RAG pipelines for grounding agents in your own data. These chapters provide the patterns you will use in virtually every agent you build.

The **Advanced Architectures** section (Chapters 7-9) covers reasoning strategies, multi-agent systems, and workflow orchestration. These chapters address the challenges of building agents that can handle complex, multi-step tasks requiring planning, collaboration, and stateful execution.

The **Engineering Practices** section (Chapters 10-12) covers testing, observability, performance optimization, and security. These are the practices that separate production-ready systems from fragile prototypes, and they are especially critical for local agents where you own the entire stack.

The **Deployment and Integration** section (Chapter 13) covers deploying agents as services, building APIs and web interfaces, integrating with desktop environments, and packaging systems for real-world use.

Each chapter includes complete, working code examples with all dependencies, project structures, and configuration files provided. You can copy the code, run it, modify it, and use it as a starting point for your own implementations. The examples follow software engineering best practices: they are modular, well-documented, handle errors, and are designed to be extended rather than replaced.

All code is written in Python 3.10+ using the official Ollama Python client library, with standard libraries and widely-used open-source packages. The implementations are compatible with current stable versions of Ollama and the referenced libraries at the time of writing. Where multiple approaches exist, alternative implementations are discussed along with their tradeoffs so you can choose what fits your context best.

The goal is not to show you every possible way to build an agent but to show you the ways that work in production, with clear reasoning about why they work and when to use them. By the end of this book, you will have a comprehensive toolkit for building local AI agents that are reliable, maintainable, and capable of solving real problems.

Key Takeaways

- Local AI agents offer significant advantages in cost, privacy, latency, and control compared to cloud APIs, especially for production workloads with consistent usage.

- An AI agent is distinguished from a simple chat interface by its ability to pursue goals, use tools, maintain state, plan and reason, and operate autonomously within defined boundaries.
- Building reliable agents requires applying software engineering rigor to handle failure modes that most tutorials ignore: malformed outputs, tool failures, context overflow, security attacks, and resource constraints.
- Local agents are a distinct engineering discipline requiring new patterns for reliability, observability, and resource management that this book establishes with complete, production-ready implementations.

Chapter 1: Foundations of Local LLMs and Ollama

Before building agents, you need to understand the infrastructure they run on. This chapter covers how local LLM inference works at a technical level, how to install and configure Ollama correctly for production use, and how to interact with it programmatically. By the end, you will have a working Ollama installation serving models via its API, with a minimal agent implementation that demonstrates the basic request-response pattern.

How Local Inference Works

When you call a cloud AI API, you send a prompt over HTTP and receive text back. The complexity of model loading, GPU management, and inference execution is hidden behind that API call. Running locally, all that complexity becomes visible, and understanding it is essential for building reliable systems.

At its core, LLM inference is a sequential computation process. The model receives a sequence of tokens (numerical representations of text), processes them through multiple layers of neural network operations, and produces probability distributions over possible next tokens at each step. The highest-probability token (or a sampled token, depending on the temperature setting) becomes the output, which is then fed back as input for the next step. This loop continues until the model generates a special end-of-sequence token or reaches a maximum length limit [1,4].

The computational characteristics of this process matter for system design:

- **Memory-bound, not compute-bound**: For most models, inference speed is limited by how fast data can be moved from memory to the processing units, not by raw computation. This is why quantization (reducing numerical precision) often improves speed as much as it reduces memory usage. Smaller weights mean less data to move [1,5,12].

- **Sequential token generation**: Each token depends on all previous tokens, making parallelization across the sequence difficult. This means inference latency scales linearly with output length, unlike training where batches can be processed in parallel.
- **KV cache overhead**: To avoid recomputing attention for previously generated tokens, models maintain a key-value cache that grows with context length. This cache can consume significant memory, especially for long conversations or documents [10].

The GGUF format is the standard for local LLM distribution. Created by Georgi Gerganov as part of the llama.cpp project, GGUF (Georgi Gerganov Universal Format) is a binary container that bundles everything needed to run a model in a single file: quantized weights, tokenizer vocabulary and rules, chat template configuration, and metadata describing the model architecture [3,4,6]. This self-contained design means you can share a GGUF file and be confident it will run on any compatible inference engine without separate configuration files or tokenizer downloads.

Quantization is how models become small enough to run on consumer hardware. Full-precision (FP16) models store each weight as a 16-bit floating-point number. A 7 billion parameter model in FP16 requires about 14 GB of memory just for weights, not counting the KV cache or activation memory. Quantization reduces precision: INT8 uses 8 bits per weight (half the memory), INT4 uses 4 bits (a quarter). Modern K-quants use mixed precision, keeping critical layers at higher precision while aggressively compressing others, achieving quality close to FP16 at a fraction of the size [2,5,7].

Ollama sits on top of llama.cpp as a runtime that abstracts away the complexity of model management and inference. It handles downloading models from its registry, loading them into memory (or VRAM), managing the KV cache, and exposing a clean REST API for your applications. You do not need to understand llama.cpp internals to use Ollama effectively, but understanding that this stack exists explains Ollama's design choices: why it uses GGUF exclusively, why quantization is central to its model naming, and why GPU support depends on the underlying llama.cpp capabilities [1,3,9].

Installing and Configuring Ollama

Ollama supports Linux, macOS, and Windows with automated installers that handle most configuration automatically. The installation process is straight-

forward, but production deployments require additional configuration beyond the defaults.

Basic Installation

On macOS and Windows, download the installer from ollama.com. Running it installs both the Ollama CLI tool and a background service that runs the inference server on port 11434 by default. On Linux, use the official install script:

```
curl -fsSL https://ollama.com/install.sh | sh
```

This script detects your system, installs dependencies, sets up a systemd service, and starts the Ollama server. After installation, verify it is running:

```
ollama --version
ollama serve
# In another terminal:
curl http://localhost:11434/api/version
```

The version endpoint should return JSON with the current Ollama version. If you see a connection refused error, the server is not running. On Linux with systemd, check its status with `systemctl status ollama`.

GPU Configuration

GPU acceleration dramatically improves inference speed, often by factors of 10 or more compared to CPU-only execution. Ollama auto-detects supported GPUs, but some platforms require additional configuration.

NVIDIA GPUs: Require CUDA compute capability 5.0 or higher (GTX 900 series or newer) and driver version 550 or later. Ollama uses the system-installed CUDA drivers; you do not need to install a separate CUDA toolkit. Verify GPU detection:

```
nvidia-smi
ollama run llama3.2
# Check logs for "using GPU" messages
```

For multi-GPU setups, use `CUDA_VISIBLE_DEVICES` to select which GPUs Ollama can access:

```
CUDA_VISIBLE_DEVICES=0,1 ollama serve
```

AMD GPUs: Use ROCm on Linux and Windows. Supported cards include Radeon RX 6000-9000 series, Ryzen AI processors, and Instinct MI series. On Linux, you may need to override GPU detection for some cards:

```
export HSA_OVERRIDE_GFX_VERSION=11.0.0
ollama serve
```

For SELinux-enabled systems, allow device access:

```
sudo setsebool container_use_devices=1
```

Apple Silicon: Metal acceleration works automatically after installation. No additional configuration is needed. Ollama leverages Apple's unified memory architecture, which means system RAM and GPU memory are shared, allowing larger models to run than would fit in discrete VRAM on equivalent systems [4,6,7].

Vulkan support: Provides an alternative for Intel and AMD GPUs on Windows and Linux when CUDA or ROCm are not available. Configure via `GGML_VK_VISIBLE_DEVICES` or disable with `OLLAMA_VULKAN=0`.

Production Configuration

The default Ollama configuration is designed for development and single-user use. Production deployments require tuning several environment variables and running behind a reverse proxy for security.

Key environment variables:

- `OLLAMA_HOST`: Bind address and port (default: 127.0.0.1:11434). Set to 0.0.0.0:11434 to accept connections from other machines, but only do this behind a reverse proxy with authentication.
- `OLLAMA_NUM_PARALLEL`: Maximum concurrent requests (default: 1 for most systems). Increase for multi-user deployments, but be aware each parallel request consumes additional memory.
- `OLLAMA_MAX_LOADED_MODELS`: Maximum models kept in memory simultaneously (default varies by system). Set to 2 or more if your application switches between models frequently.
- `OLLAMA_KEEP_ALIVE`: How long to keep unloaded models in memory after last use (default: 5 minutes). Increase for applications with bursty usage patterns.
- `OLLAMA_CONTEXT_LENGTH`: Default context window size (default: typically 4096-8192 depending on model). Increasing this consumes more KV cache memory.
- `OLLAMA_GPU_OVERHEAD`: Bytes of VRAM to reserve for non-model operations. Useful for preventing out-of-memory errors when running large models near GPU capacity [3,5,10].

A production systemd service configuration for Linux:

```
# /etc/systemd/system/ollama.service
[Unit]
Description=Ollama Service
After=network-online.target

[Service]
ExecStart=/usr/local/bin/ollama serve
Environment="OLLAMA_HOST=127.0.0.1:11434"
Environment="OLLAMA_NUM_PARALLEL=4"
Environment="OLLAMA_MAX_LOADED_MODELS=2"
Environment="OLLAMA_KEEP_ALIVE=24h"
Restart=always
RestartSec=5
User=ollama
Group=ollama

[Install]
WantedBy=multi-user.target
```

Docker Deployment

Docker is the standard approach for production Ollama deployments, providing isolation, reproducibility, and easy GPU passthrough configuration. The official image is `ollama/ollama`.

Basic Docker Compose for development:

```yaml
# docker-compose.yml
version: "3.8"
services:
  ollama:
    image: ollama/ollama:latest
    container_name: ollama
    restart: unless-stopped
    volumes:
      - ollama_data:/root/.ollama
    ports:
      - "11434:11434"

volumes:
  ollama_data:
```

Production Docker Compose with GPU passthrough, resource limits, and health checks:

```yaml
# docker-compose.prod.yml
version: "3.8"
services:
  ollama:
    image: ollama/ollama:latest
    container_name: ollama-prod
    restart: unless-stopped
    deploy:
      resources:
        limits:
          memory: 32G
          cpus: "8"
        reservations:
          memory: 16G
          cpus: "4"
          devices:
            - driver: nvidia
              count: all
              capabilities: [gpu]
    volumes:
```

```
      - ollama_data:/root/.ollama
    ports:
      - "127.0.0.1:11434:11434"
    environment:
      - OLLAMA_HOST=0.0.0.0:11434
      - OLLAMA_NUM_PARALLEL=4
      - OLLAMA_MAX_LOADED_MODELS=2
      - OLLAMA_KEEP_ALIVE=24h
    healthcheck:
      test: ["CMD", "curl", "-f", "http://localhost:11434/api/version"]
      interval: 30s
      timeout: 10s
      retries: 3
      start_period: 60s

volumes:
  ollama_data:
```

Critical security note: Ollama has no built-in authentication. Binding to 127.0.0.1 (localhost only) is mandatory unless you place a reverse proxy with authentication in front of it. Exposing port 11434 directly to the internet allows anyone to run inference on your GPU, pull arbitrary models, and potentially exploit vulnerabilities [3,5,6].

Understanding the Ollama API

Ollama exposes a RESTful JSON API that all client interactions use, including the CLI tool and official language libraries. Understanding the API directly is valuable even when using client libraries, because it helps you debug issues, understand latency characteristics, and design integrations that work correctly under load.

The API base URL is `http://localhost:11434/api` by default. All endpoints stream JSON by default for real-time token output; set `"stream": false` for non-streaming responses where you wait for the complete result. Model names use the format `model:tag`, with `latest` as the default tag (e.g., `llama3.2:latest` is equivalent to `llama3.2`).

Core Endpoints

POST /api/generate: Text completion for raw prompts without chat history. Useful for simple text generation tasks but less common for agents, which typically use the chat endpoint.

```
curl http://localhost:11434/api/generate -d '{
  "model": "llama3.2",
  "prompt": "Explain quantum computing in one sentence.",
  "stream": false
}'
```

POST /api/chat: Chat completions with message history and tool support. This is the primary endpoint for agent interactions. Messages use role-based formatting: `system` for instructions, `user` for input, `assistant` for model responses, and `tool` for tool execution results [1,2].

```
curl http://localhost:11434/api/chat -d '{
  "model": "llama3.2",
  "messages": [
    {"role": "system", "content": "You are a helpful coding assistant."},
    {"role": "user", "content": "Write a Python function to calculate fibonacci
↪  numbers."}
  ],
  "stream": false
}'
```

POST /api/embed: Generate text embeddings for RAG and similarity search. Takes the embedding model name and input text, returns a vector of floating-point numbers [2].

```
curl http://localhost:11434/api/embed -d '{
  "model": "nomic-embed-text",
  "input": "This is a sample document about local AI agents."
}'
```

POST /api/pull: Download a model from the Ollama registry. Supports streaming progress updates.

```
curl -s http://localhost:11434/api/pull -d '{"name": "llama3.2"}'
```

GET /api/tags: List all locally available models with their sizes and IDs.

```
curl http://localhost:11434/api/tags
```

GET /api/ps: List currently loaded (in-memory) models. Useful for monitoring resource usage and understanding which models are actively consuming GPU memory [1].

```
curl http://localhost:11434/api/ps
```

Streaming Responses

Streaming is fundamental to good agent UX because it lets users see responses as they are generated rather than waiting for completion. The API uses Server-Sent Events format: each chunk is a JSON object followed by a newline, containing either a partial response or metadata about token usage and timing.

Example streaming request:

```
curl -N http://localhost:11434/api/chat -d '{
  "model": "llama3.2",
  "messages": [{"role": "user", "content": "Count from 1 to 5."}],
  "stream": true
}'
```

Each chunk contains fields like `message.content` (partial text), `done` (boolean indicating completion), and timing information. Your application accumulates chunks until `done` is true, then processes the complete response.

OpenAI-Compatible Endpoint

Ollama provides an OpenAI-compatible API at `http://localhost:11434/v1` that supports the same request/response format as OpenAI's chat completions endpoint. This enables using Ollama with any library or framework designed for OpenAI, including LangChain, LlamaIndex, and many agent frameworks [2].

```
curl http://localhost:11434/v1/chat/completions -H "Content-Type:
↪  application/json" -d '{
  "model": "llama3.2",
  "messages": [{"role": "user", "content": "Hello!"}],
  "stream": false
}'
```

This compatibility layer is valuable for framework integration but has some limitations: not all OpenAI-specific features are supported, and the tool calling format follows Ollama's native conventions rather than exact OpenAI parity. For production agents, using the official Ollama Python client directly is recommended over relying on OpenAI compatibility.

Model Formats and GGUF

Understanding GGUF is important for model selection, custom model creation, and troubleshooting. The format has become the de facto standard for local LLM deployment because it solves real problems that earlier formats did not address well [3,4,6].

Structure of a GGUF File

A GGUF file contains four main components:

1. **Header**: Version number and metadata about the file format.
2. **Tensor data**: The model weights, optionally quantized to reduce size.
3. **Tokenizer configuration**: Vocabulary and rules for converting text to tokens and back.
4. **Metadata**: Model architecture parameters, chat template, author information, and other attributes.

The format is designed to be memory-mapped directly into process memory without parsing overhead, which speeds up model loading significantly. The flexible key-value metadata system allows the format to evolve without breaking compatibility with older models [4,19].

Quantization Types

GGUF supports multiple quantization methods, each trading quality for size. Understanding these tradeoffs is essential for choosing the right model variant for your hardware and task requirements.

Common quantization types:

- **Q4_K_M**: 4-bit K-quant medium. The default recommendation for most use cases. Achieves approximately 97.5% of FP16 quality at roughly 25% of the memory footprint. Uses mixed precision, keeping attention layers at higher bit depth while compressing feed-forward layers more aggressively [2,5,7].

- **Q5_K_M**: 5-bit K-quant medium. Better quality than Q4_K_M (approximately 98.5% of FP16) at the cost of about 10-15% more file size. Worth using when you have extra VRAM and need the best quality from a given model architecture [4,10].
- **Q6_K**: 6-bit K-quant. Very close to Q8_0 in quality but slightly smaller. Good choice for 16GB+ VRAM systems that want near-lossless performance without full 8-bit overhead [5].
- **Q8_0**: 8-bit quantization. Effectively lossless for most practical purposes, within 0.1-0.5% of FP16 on standard benchmarks. Use when memory is not a constraint and you want maximum fidelity [2,5,7].
- **IQ4_NL / IQ4_XS**: Improved 4-bit variants with different outlier handling strategies. IQ4_NL (no linear) can outperform Q4_K_M on some models while using similar memory [2].

The quality cliff is between Q3 and Q4 quantization levels, not between Q4 and Q8. Going from Q3 to Q4 typically yields a noticeable improvement in output quality, while going from Q4 to Q8 often produces changes that are difficult to perceive in practical use [3,7]. This means Q4_K_M is usually the smartest choice: you get most of the quality at a fraction of the cost.

Custom Models with Modelfile

Ollama supports creating custom models using a Modelfile, which specifies the base model, system prompt, parameters, and other configuration. This is useful for tailoring models to specific tasks without fine-tuning [1].

Example Modelfile:

```
# Modelfile
FROM llama3.2

SYSTEM """You are a senior software engineer specializing in Python.
Always provide complete, runnable code examples.
Explain your reasoning briefly before each code block."""

PARAMETER temperature 0.3
PARAMETER num_ctx 8192
```

Build and run:

```
ollama create my-coding-assistant -f Modelfile
ollama run my-coding-assistant "Write a FastAPI endpoint that validates email
↪  addresses."
```

Modelfiles can also be used to customize parameters like temperature, top_p, and context length for specific use cases. The `SYSTEM` directive sets the system prompt that is prepended to every conversation, providing consistent behavioral guidance.

First Agent: A Minimal Example

Now that Ollama is installed and configured, let us build a minimal agent to demonstrate the basic interaction pattern. This example is intentionally simple: it takes a user query, sends it to the model with a system prompt, and returns the response. Later chapters will add tool calling, memory, planning, and other capabilities on top of this foundation.

Project Structure

Create a new directory for your project:

```
mkdir ollama-agent
cd ollama-agent
```

Create a requirements file:

```
# requirements.txt
ollama>=0.4.0
pydantic>=2.0.0
httpx>=0.27.0
```

Install dependencies:

```
pip install -r requirements.txt
```

The Minimal Agent

Create `agent.py`:

```python
"""Minimal Ollama agent demonstrating basic request-response pattern."""

import ollama
from dataclasses import dataclass


@dataclass
class AgentConfig:
    """Configuration for the agent."""
    model: str = "llama3.2"
    system_prompt: str = (
        "You are a helpful AI assistant. "
        "Provide clear, concise, and accurate responses. "
        "If you do not know the answer, say so honestly."
    )
    temperature: float = 0.7


class MinimalAgent:
    """A minimal agent that sends prompts to Ollama and returns responses."""

    def __init__(self, config: AgentConfig | None = None):
        self.config = config or AgentConfig()

    def chat(self, user_message: str) -> str:
        """Send a single message and get the model response."""
        messages = [
            {"role": "system", "content": self.config.system_prompt},
            {"role": "user", "content": user_message},
        ]

        response = ollama.chat(
            model=self.config.model,
            messages=messages,
            options={"temperature": self.config.temperature},
        )

        return response["message"]["content"]

    def chat_stream(self, user_message: str):
        """Stream the model response token by token."""
        messages = [
            {"role": "system", "content": self.config.system_prompt},
            {"role": "user", "content": user_message},
        ]

        response = ollama.chat(
            model=self.config.model,
            messages=messages,
            stream=True,
```

```
            options={"temperature": self.config.temperature},
        )

        for chunk in response:
            yield chunk["message"]["content"]


def main():
    """Interactive demo of the minimal agent."""
    config = AgentConfig()
    agent = MinimalAgent(config)

    print(f"Agent initialized with model: {config.model}")
    print("Type 'quit' to exit.\n")

    while True:
        user_input = input("You: ").strip()
        if user_input.lower() in ("quit", "exit", "q"):
            print("Goodbye!")
            break

        if not user_input:
            continue

        print("Agent: ", end="", flush=True)
        for token in agent.chat_stream(user_input):
            print(token, end="", flush=True)
        print()  # Newline after response


if __name__ == "__main__":
    main()
```

How It Works, Line by Line

The `AgentConfig` dataclass centralizes configuration: model name, system prompt, and temperature. This pattern makes it easy to swap models or adjust behavior without changing the core logic. The system prompt defines the agent's personality and constraints; even in this minimal example, it establishes expectations about response quality and honesty.

The `MinimalAgent.chat` method constructs a message list with the system prompt and user input, then calls `ollama.chat()`. The response is a dictionary containing the model's reply in `response["message"]["content"]`. This is the fundamental pattern you will see throughout this book: construct messages, call the API, extract the response.

The `chat_stream` method uses `stream=True` to receive tokens as they are generated. It yields each chunk of content, allowing the caller to display text in real time. Streaming is essential for good UX in interactive applications because it reduces perceived latency: users see the first words almost immediately rather than waiting for the complete response.

The `main` function implements a simple REPL (read-eval-print loop) that lets you interact with the agent directly from the terminal. It uses the streaming method so responses appear as they are generated, mimicking the experience of chatting with any modern AI assistant.

Running and Testing

Before running, ensure Ollama is serving and has the model downloaded:

```
ollama serve
# In another terminal:
ollama pull llama3.2
```

Run the agent:

```
python agent.py
```

Expected output:

```
Agent initialized with model: llama3.2
Type 'quit' to exit.

You: What is the capital of France?
Agent: The capital of France is Paris.

You: Write a Python function to reverse a string.
Agent: Here is a simple function to reverse a string:

def reverse_string(s: str) -> str:
    return s[::-1]

# Example usage:
print(reverse_string("hello"))  # Output: "olleh"

You: quit
Goodbye!
```

Common Pitfalls

- **Connection refused**: Ollama server is not running. Start it with `ollama serve` or ensure the systemd service is active on Linux.
- **Model not found**: The model has not been pulled yet. Run `ollama pull <model-name>` before using it in your code.
- **Slow responses on first run**: The model must be loaded into memory before generating tokens. First requests are slow; subsequent requests to the same model are fast because the model stays loaded for a keep-alive period.
- **Out of memory errors**: The model is too large for available RAM/VRAM. Use a smaller model or a lower quantization level (e.g., `llama3.2:8b-q4_K_M` instead of the default).
- **Streaming appears to hang**: If you use non-streaming mode (`stream=False`) with a large model and long response, the request may take many seconds before returning anything. Always prefer streaming for interactive applications.

Key Takeaways

- Local LLM inference is memory-bound and sequential, which has important implications for performance optimization and resource management in agent systems.
- GGUF is the standard format for local models, bundling weights, tokenizer, and metadata into a single portable file with support for multiple quantization levels.
- Q4_K_M quantization offers the best balance of quality and size for most use cases, achieving approximately 97.5% of FP16 quality at roughly 25% of the memory footprint.
- Ollama provides a clean REST API with endpoints for chat, generation, embeddings, and model management, plus OpenAI-compatible endpoints for framework integration.
- Production Ollama deployments require explicit configuration: resource limits, parallel request handling, GPU passthrough in containers, and a reverse proxy for authentication since Ollama has no built-in security.
- The basic agent pattern is simple: construct messages with system prompt and user input, call the API, extract the response. All advanced agent capabilities build on this foundation.

Chapter 2: Choosing and Managing Models

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/buildingaiagentswithollama.

The Local Model Landscape

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/buildingaiagentswithollama.

Model Families and Their Strengths

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/buildingaiagentswithollama.

Benchmark Interpretation

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/buildingaiagentswithollama.

Hardware Requirements by Model Size

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/buildingaiagentswithollama.

Performance vs Size Tradeoffs

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/buildingaiagentswithollama.

Speed Considerations

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/buildingaiagentswithollama.

Capability vs Task Complexity

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/buildingaiagentswithollama.

Cost Analysis

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/buildingaiagentswithollama.

Quantization Explained

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/buildingaiagentswithollama.

How Quantization Works

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/buildingaiagentswithollama.

Quantization Quality Comparison

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/buildingaiagentswithollama.

When to Use Higher Quantization

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/buildingaiagentswithollama.

Multi-Model Strategies

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/buildingaiagentswithollama.

Task-Based Model Selection

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/buildingaiagentswithollama.

Model Registry Implementation

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/buildingaiagentswithollama.

Model Routing in Agents

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/buildingaiagentswithollama.

Key Takeaways

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/buildingaiagentswithollama.

Chapter 3: Prompt Engineering for Agents

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/buildingaiagentswithollama.

From Prompts to Specifications

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/buildingaiagentswithollama.

The Specification Mindset

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/buildingaiagentswithollama.

System Prompt Architecture

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/buildingaiagentswithollama.

Structured Prompt Design Patterns

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/buildingaiagentswithollama.

Role-Based Prompting

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/buildingaiagentswithollama.

Constraint-Based Prompting

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/buildingaiagentswithollama.

Step-by-Step Decomposition

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/buildingaiagentswithollama.

Guardrails and Safety Instructions

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/buildingaiagentswithollama.

System Prompts and Role Definitions

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/buildingaiagentswithollama.

The Layered System Prompt

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/buildingaiagentswithollama.

Dynamic System Prompts

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/buildingaiagentswithollama.

Few-Shot and Example-Based Prompting

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/buildingaiagentswithollama.

When to Use Few-Shot Prompting

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/buildingaiagentswithollama.

Example Implementation

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/buildingaiagentswithollama.

Best Practices for Few-Shot Examples

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/buildingaiagentswithollama.

Testing and Iterating Prompts

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/buildingaiagentswithollama.

Prompt Versioning

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/buildingaiagentswithollama.

Prompt Evaluation Framework

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/buildingaiagentswithollama.

Iteration Workflow

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/buildingaiagentswithollama.

Key Takeaways

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/buildingaiagentswithollama.

Chapter 4: Structured Outputs and Tool Calling

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/buildingaiagentswithollama.

JSON Mode and Schema Enforcement

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/buildingaiagentswithollama.

Ollama's Format Parameter

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/buildingaiagentswithollama.

Pydantic Models for Type Safety

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/buildingaiagentswithollama.

Basic Structured Output with Pydantic

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/buildingaiagentswithollama.

How It Works

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/buildingaiagentswithollama.

Function Calling Patterns

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/buildingaiagentswithollama.

Ollama's Native Tool Calling

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/buildingaiagentswithollama.

How ToolCallingAgent Works

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/buildingaiagentswithollama.

Python SDK Function-as-Tool Shortcut

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/buildingaiagentswithollama.

Multi-Tool Selection and Parallel Calls

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/buildingaiagentswithollama.

Error Recovery for Bad Outputs

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/buildingaiagentswithollama.

Structured Output Recovery Strategies

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/buildingaiagentswithollama.

Tool Calling Error Handling

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/buildingaiagentswithollama.

How RobustToolAgent Works

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/buildingaiagentswithollama.

Building a Tool Registry

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/buildingaiagentswithollama.

Browser Automation Tools

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/buildingaiagentswithollama.

Why Playwright Over Selenium

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/buildingaiagentswithollama.

Installation

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/buildingaiagentswithollama.

Basic Browser Control Tool

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/buildingaiagentswithollama.

How It Works

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/buildingaiagentswithollama.

Integrating Browser Tools with an Agent

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/buildingaiagentswithollama.

Security Considerations for Browser Automation

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/buildingaiagentswithollama.

Common Pitfalls

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/buildingaiagentswithollama.

Key Takeaways

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/buildingaiagentswithollama.

Chapter 5: Memory Systems and Context Management

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/buildingaiagentswithollama.

Chapter Dependencies and Project Structure

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/buildingaiagentswithollama.

The Memory Problem

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/buildingaiagentswithollama.

Short-Term Conversation Memory

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/buildingaiagentswithollama.

Basic Message History

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/buildingaiagentswithollama.

Sliding Window Memory

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/buildingaiagentswithollama.

Summarization-Based Memory

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/buildingaiagentswithollama.

How SummarizingMemory Works

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/buildingaiagentswithollama.

Tradeoffs to understand:

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/buildingaiagentswithollama.

Long-Term Memory Architectures

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/buildingaiagentswithollama.

Types of Long-Term Memory

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/buildingaiagentswithollama.

Vector-Based Long-Term Memory

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/buildingaiagentswithollama.

How LongTermMemory Works

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/buildingaiagentswithollama.

Key design decisions:

Context Window Optimization

Token Budgeting

Selective Context Inclusion

Context Compression

Implementing a Memory Manager

Key Takeaways

Chapter 6: Retrieval-Augmented Generation and Vector Search

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/buildingaiagentswithollama.

Chapter Dependencies and Project Structure

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/buildingaiagentswithollama.

Why RAG Is Essential for Agents

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/buildingaiagentswithollama.

Embeddings and Vector Spaces

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/buildingaiagentswithollama.

How Embeddings Work

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/buildingaiagentswithollama.

Choosing an Embedding Model

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/buildingaiagentswithollama.

Generating Embeddings with Ollama

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/buildingaiagentswithollama.

Vector Database Options

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/buildingaiagentswithollama.

Lightweight (Development and Small Scale)

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/buildingaiagentswithollama.

Production-Grade (Self-Hosted)

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/buildingaiagentswithollama.

Database Extensions

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/buildingaiagentswithollama.

Recommendation Matrix

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/buildingaiagentswithollama.

Document Ingestion Pipelines

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/buildingaiagentswithollama.

Pipeline Architecture

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/buildingaiagentswithollama.

Complete Ingestion Implementation

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/buildingaiagentswithollama.

Chunking Strategy

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/buildingaiagentswithollama.

Hybrid Search and Re-Ranking

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/buildingaiagentswithollama.

Why Hybrid Search Matters

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/buildingaiagentswithollama.

Re-Ranking for Quality

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/buildingaiagentswithollama.

Complete RAG Query Implementation

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/buildingaiagentswithollama.

How RAGQueryEngine Works

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/buildingaiagentswithollama.

Key design decisions:

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/buildingaiagentswithollama.

RAG Failure Modes: Retrieval Drift and Degradation

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/buildingaiagentswithollama.

Key Takeaways

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/buildingaiagentswithollama.

Chapter 7: Planning, Reasoning, and Complex Tasks

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/buildingaiagentswithollama.

Beyond Single-Turn Responses

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/buildingaiagentswithollama.

Chain of Thought and Reasoning

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/buildingaiagentswithollama.

How Chain of Thought Works

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/buildingaiagentswithollama.

Implementing Reasoning Agents

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/buildingaiagentswithollama.

When Chain of Thought Matters Most

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/buildingaiagentswithollama.

The ReAct Pattern

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/buildingaiagentswithollama.

ReAct Loop Mechanics

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/buildingaiagentswithollama.

Implementing a ReAct Agent

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/buildingaiagentswithollama.

ReAct vs Direct Tool Calling

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/buildingaiagentswithollama.

Task Decomposition Strategies

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/buildingaiagentswithollama.

Hierarchical Decomposition

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/buildingaiagentswithollama.

When to Use Decomposition

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/buildingaiagentswithollama.

Key Takeaways

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/buildingaiagentswithollama.

Chapter 8: Multi-Agent Architectures

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/buildingaiagentswithollama.

Why Multiple Agents

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/buildingaiagentswithollama.

Specialization and Roles

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/buildingaiagentswithollama.

Separation of Concerns

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/buildingaiagentswithollama.

Debate and Quality Improvement

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/buildingaiagentswithollama.

Specialization and Roles

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/buildingaiagentswithollama.

Common Agent Roles

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/buildingaiagentswithollama.

Communication Patterns

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/buildingaiagentswithollama.

Sequential Chain

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/buildingaiagentswithollama.

Parallel Fan-Out

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/buildingaiagentswithollama.

Hierarchical (Supervisor)

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/buildingaiagentswithollama.

Debate Pattern

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/buildingaiagentswithollama.

Orchestration Strategies

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/buildingaiagentswithollama.

Centralized Orchestration

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/buildingaiagentswithollama.

Event-Driven Orchestration

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/buildingaiagentswithollama.

Graph-Based Orchestration

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/buildingaiagentswithollama.

Building an Agent Team

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/buildingaiagentswithollama.

Key Takeaways

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/buildingaiagentswithollama.

Chapter 9: Workflows, State Machines, and Orchestration

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/buildingaiagentswithollama.

Agents as Workflow Nodes

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/buildingaiagentswithollama.

Workflow Components

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/buildingaiagentswithollama.

Simple Workflow Engine

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/buildingaiagentswithollama.

State Machine Design

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/buildingaiagentswithollama.

State Machine Pattern

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/buildingaiagentswithollama.

How AgentStateMachine Works

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/buildingaiagentswithollama.

Async Processing and Concurrency

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/buildingaiagentswithollama.

Python asyncio for Agent Workflows

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/buildingaiagentswithollama.

Retries, Timeouts, and Circuit Breakers

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/buildingaiagentswithollama.

Retry with Exponential Backoff

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/buildingaiagentswithollama.

Timeouts

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/buildingaiagentswithollama.

Circuit Breaker Pattern

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/buildingaiagentswithollama.

Key Takeaways

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/buildingaiagentswithollama.

Chapter 10: Testing, Debugging, and Observability

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/buildingaiagentswithollama.

Chapter Dependencies and Project Structure

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/buildingaiagentswithollama.

The Testing Challenge

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/buildingaiagentswithollama.

Unit Testing Agents and Tools

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/buildingaiagentswithollama.

Testing Prompt Templates

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/buildingaiagentswithollama.

Testing Tool Functions

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/buildingaiagentswithollama.

Testing Output Parsers

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/buildingaiagentswithollama.

Integration and End-to-End Tests

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/buildingaiagentswithollama.

Golden Dataset Testing

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/buildingaiagentswithollama.

LLM-as-Judge Evaluation

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/buildingaiagentswithollama.

Logging and Tracing

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/buildingaiagentswithollama.

Structured Logging

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/buildingaiagentswithollama.

Execution Tracing

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/buildingaiagentswithollama.

Monitoring and Alerting

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/buildingaiagentswithollama.

Core Metrics to Monitor

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/buildingaiagentswithollama.

Implementation with Prometheus (Example)

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/buildingaiagentswithollama.

Key Takeaways

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/buildingaiagentswithollama.

Chapter 11: Performance, Resources, and Optimization

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/buildingaiagentswithollama.

Understanding Resource Consumption

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/buildingaiagentswithollama.

Model Weights

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/buildingaiagentswithollama.

KV Cache

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/buildingaiagentswithollama.

Compute

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/buildingaiagentswithollama.

GPU Acceleration and Configuration

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/buildingaiagentswithollama.

Verifying GPU Usage

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/buildingaiagentswithollama.

NVIDIA CUDA Configuration

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/buildingaiagentswithollama.

AMD ROCm Configuration

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/buildingaiagentswithollama.

Apple Metal Configuration

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/buildingaiagentswithollama.

GPU Memory Management

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/buildingaiagentswithollama.

Caching Strategies

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/buildingaiagentswithollama.

Prompt-Response Caching

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/buildingaiagentswithollama.

Semantic Caching

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/buildingaiagentswithollama.

Batch Processing and Throughput

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/buildingaiagentswithollama.

Request Batching

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/buildingaiagentswithollama.

Async Batch Processing

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/buildingaiagentswithollama.

Scaling Local Inference

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/buildingaiagentswithollama.

Horizontal Scaling with Load Balancing

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/buildingaiagentswithollama.

Kubernetes Deployment

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/buildingaiagentswithollama.

Model-Level Scaling

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/buildingaiagentswithollama.

GPU Fragmentation and Memory Pressure Scenarios

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/buildingaiagentswithollama.

Key Takeaways

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/buildingaiagentswithollama.

Chapter 12: Security, Privacy, and Sandboxing

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/buildingaiagentswithollama.

The Threat Model for Local Agents

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/buildingaiagentswithollama.

Attack Surface Analysis

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/buildingaiagentswithollama.

OWASP Top Risks for LLM Applications

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/buildingaiagentswithollama.

Prompt Injection Attacks

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/buildingaiagentswithollama.

Direct Prompt Injection

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/buildingaiagentswithollama.

Indirect Prompt Injection

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/buildingaiagentswithollama.

Defense Strategies

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/buildingaiagentswithollama.

Tool Sandboxing

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/buildingaiagentswithollama.

Principle of Least Privilege

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/buildingaiagentswithollama.

Sandboxing Code Execution

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/buildingaiagentswithollama.

Network Access Control

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/buildingaiagentswithollama.

Data Privacy and Compliance

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/buildingaiagentswithollama.

Data Minimization

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/buildingaiagentswithollama.

Access Control

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/buildingaiagentswithollama.

Compliance Considerations

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/buildingaiagentswithollama.

Hardening Your Deployment

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/buildingaiagentswithollama.

Network Security

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/buildingaiagentswithollama.

Container Security

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/buildingaiagentswithollama.

API Security

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/buildingaiagentswithollama.

Model Security

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/buildingaiagentswithollama.

Key Takeaways

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/buildingaiagentswithollama.

Chapter 13: Deployment, APIs, and Real-World Integration

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/buildingaiagentswithollama.

Chapter Dependencies and Project Structure

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/buildingaiagentswithollama.

Packaging Agents as Services

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/buildingaiagentswithollama.

FastAPI Agent Service

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/buildingaiagentswithollama.

RESTful Agent APIs

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/buildingaiagentswithollama.

API Design Principles

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/buildingaiagentswithollama.

Conversation Management

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/buildingaiagentswithollama.

Web Interfaces and Dashboards

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/buildingaiagentswithollama.

Simple Chat Interface with HTML/JS

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/buildingaiagentswithollama.

Desktop and System Integration

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/buildingaiagentswithollama.

File System Agent

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/buildingaiagentswithollama.

Automation Integration

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/buildingaiagentswithollama.

Production Deployment Patterns

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/buildingaiagentswithollama.

Complete Docker Stack

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/buildingaiagentswithollama.

Deployment Checklist

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/buildingaiagentswithollama.

Key Takeaways

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/buildingaiagentswithollama.

Chapter 14: Capstone Project — Building a Production Coding Agent

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/buildingaiagentswithollama.

Project Overview: The CodePilot Agent

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/buildingaiagentswithollama.

Project Structure

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/buildingaiagentswithollama.

Step 1: Configuration and Dependencies

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/buildingaiagentswithollama.

Step 2: Tool Registry and Implementation

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/buildingaiagentswithollama.

Step 3: Memory System Integration

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/buildingaiagentswithollama.

Step 4: RAG Pipeline for Codebase Indexing

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/buildingaiagentswithollama.

Step 5: Agent Core with Prompts and Reasoning

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/buildingaiagentswithollama.

How CodePilotAgent Works

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/buildingaiagentswithollama.

Step 6: Security Layer

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/buildingaiagentswithollama.

Step 7: Observability Setup

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/buildingaiagentswithollama.

Step 8: FastAPI Service Layer

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/buildingaiagentswithollama.

Step 9: Docker Deployment

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/buildingaiagentswithollama.

Step 10: Web Interface

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/buildingaiagentswithollama.

Step 11: Testing

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/buildingaiagentswithollama.

Running the Complete System

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/buildingaiagentswithollama.

Architecture Summary

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/buildingaiagentswithollama.

Key Takeaways

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/buildingaiagentswithollama.

Conclusion: The Future of Local Agents

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/buildingaiagentswithollama.

What We Have Built

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/buildingaiagentswithollama.

The Trajectory Ahead

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/buildingaiagentswithollama.

Principles That Will Endure

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/buildingaiagentswithollama.

Key Takeaways

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/buildingaiagentswithollama.

References

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/buildingaiagentswithollama.

Appendix A: Ollama Environment Variables and Configuration Reference

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/buildingaiagentswithollama.

Network and Binding Variables

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/buildingaiagentswithollama.

Model Storage Variables

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/buildingaiagentswithollama.

Concurrency and Queue Variables

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/buildingaiagentswithollama.

GPU and Hardware Variables

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/buildingaiagentswithollama.

Context and Inference Variables

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/buildingaiagentswithollama.

Setting Environment Variables by Platform

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/buildingaiagentswithollama.

Recommended Production Configurations

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/buildingaiagentswithollama.

Appendix B: Troubleshooting Common GPU, Driver, and OOM Errors

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/buildingaiagentswithollama.

Enabling Debug Logging

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/buildingaiagentswithollama.

NVIDIA GPU Not Detected

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/buildingaiagentswithollama.

AMD GPU Not Detected

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/buildingaiagentswithollama.

CUDA Out of Memory (OOM) Errors

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/buildingaiagentswithollama.

CPU Fallback (Silent or Explicit)

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/buildingaiagentswithollama.

Model Loading Fails or Hangs

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/buildingaiagentswithollama.

Connection Refused Errors

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/buildingaiagentswithollama.

Apple Silicon Specific Issues

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/buildingaiagentswithollama.

Diagnostic Checklist

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/buildingaiagentswithollama.

Appendix C: Model Quantization Cheat Sheet

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/buildingaiagentswithollama.

Quick Decision Guide

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/buildingaiagentswithollama.

Quantization Level Comparison

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/buildingaiagentswithollama.

Key Principles

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/buildingaiagentswithollama.

VRAM Estimation Formula

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/buildingaiagentswithollama.

Model Tag Naming Convention

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/buildingaiagentswithollama.

Speed vs Quantization

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/buildingaiagentswithollama.

Appendix D: Reusable Code Patterns

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/buildingaiagentswithollama.

Pattern 1: Retry with Exponential Backoff and Jitter

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/buildingaiagentswithollama.

Pattern 2: Circuit Breaker for External Services

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/buildingaiagentswithollama.

Pattern 3: Structured JSON Logger

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/buildingaiagentswithollama.

Pattern 4: Token Counter Utility

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/buildingaiagentswithollama.

Pattern 5: Rate Limiter for API Calls

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/buildingaiagentswithollama.

Pattern 6: Context Window Manager

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/buildingaiagentswithollama.