I taught myself how to build RAG + AI Agents in production.  
Been running them live for over a year now. Here are 4 steps + the only resources you really need to do the same.  
  
Ugly truth: most “AI Engineers” shouting on social media haven’t built a single real production AI Agent or RAG system.  
If you want to be different - actually build and ship these systems: here’s a laser-focused roadmap from my own journey.  
  
🚀 𝗦𝘁𝗮𝗿𝘁 𝘄𝗶𝘁𝗵 𝗳𝘂𝗻𝗱𝗮𝗺𝗲𝗻𝘁𝗮𝗹𝘀  
Because no matter how fast LLM/GenAI evolves, your ML & software foundations keep you relevant.  
  
✅ Hands-On ML with TensorFlow & Keras: [**https://lnkd.in/dWrf5pbS**](https://lnkd.in/dWrf5pbS)  
✅ ISLR: [**https://lnkd.in/djGPVVwJ**](https://lnkd.in/djGPVVwJ)  
✅ Machine Learning for Beginners by Microsoft (free curriculum):  
[**https://lnkd.in/d8kZA3es**](https://lnkd.in/d8kZA3es)  
  
1️⃣ 𝗠𝗮𝘀𝘁𝗲𝗿 𝗟𝗟𝗠𝘀 & 𝗚𝗲𝗻𝗔𝗜 𝗦𝘆𝘀𝘁𝗲𝗺𝘀  
→ Learn to build & deploy LLMs, understand system design tradeoffs, and handle real constraints.  
  
📚 Must-reads:  
✅ Designing ML Systems – Chip Huyen: [**https://lnkd.in/guN-UhXA**](https://lnkd.in/guN-UhXA)  
✅ The LLM Engineering Handbook – Iusztin & Labonne: [**https://lnkd.in/gyA4vFXz**](https://lnkd.in/gyA4vFXz)  
✅ Build a LLM (From Scratch) – Raschka: [**https://lnkd.in/gXNa-SPb**](https://lnkd.in/gXNa-SPb)  
✅ Hands-On LLMs GitHub: [**https://lnkd.in/eV4qrgNW**](https://lnkd.in/eV4qrgNW)  
  
2️⃣ 𝗚𝗼 𝗯𝗲𝘆𝗼𝗻𝗱 𝘁𝗵𝗲 𝗵𝘆𝗽𝗲 𝗼𝗻 𝗔𝗜 𝗔𝗴𝗲𝗻𝘁𝘀  
→ Most demos = “if user says hello, return hello.”  
Actual agents? Handle memory, tools, workflows, costs.  
  
✅ AI Agents for Beginners (GitHub): [**https://lnkd.in/eik2btmq**](https://lnkd.in/eik2btmq)  
✅ GenAI Agents – build step by step: [**https://lnkd.in/dnhwk75V**](https://lnkd.in/dnhwk75V)  
✅ OpenAI’s guide to agents: [**https://lnkd.in/guRfXsFK**](https://lnkd.in/guRfXsFK)  
✅ Anthropic’s Building Effective Agents: [**https://lnkd.in/gRWKANS4**](https://lnkd.in/gRWKANS4)  
  
3️⃣ 𝗥𝗔𝗚 𝗶𝘀 𝗻𝗼𝘁 𝗷𝘂𝘀𝘁 𝗮 𝘃𝗲𝗰𝘁𝗼𝗿 𝗗𝗕  
Real Retrieval-Augmented Generation requires:  
→ Chunking, hybrid BM25 + vectors, reranking  
→ Query routing & fallback  
→ Evaluating retrieval quality, not just LLM output  
  
✅ RAG Techniques repo: [**https://lnkd.in/dD4S8Cq2**](https://lnkd.in/dD4S8Cq2)  
✅ Advanced RAG: [**https://lnkd.in/g2ZHwZ3w**](https://lnkd.in/g2ZHwZ3w)  
✅ Cost-efficient retrieval with Postgres/OpenSearch/Qdrant  
✅ Monitoring with Langfuse / Comet  
  
4️⃣ 𝗚𝗲𝘁 𝘀𝗲𝗿𝗶𝗼𝘂𝘀 𝗼𝗻 𝗦𝗼𝗳𝘁𝘄𝗮𝗿𝗲 & 𝗜𝗻𝗳𝗿𝗮  
→ FastAPI, async Python, Pydantic  
→ Docker, CI/CD, blue-green deploys  
→ ETL orchestration (Airflow, Step Functions)  
→ Logs + metrics (CloudWatch, Prometheus)  
  
✅ Move to production: [**https://lnkd.in/dnnkrJbE**](https://lnkd.in/dnnkrJbE)  
✅ Made with ML (full ML+infra): [**https://lnkd.in/e-XQwXqS**](https://lnkd.in/e-XQwXqS)  
✅ AWS GenAI path: [**https://lnkd.in/dmhR3uPc**](https://lnkd.in/dmhR3uPc)  
  
5️⃣ 𝗪𝗵𝗲𝗿𝗲 𝗱𝗼 𝗜 𝗹𝗲𝗮𝗿𝗻 𝗳𝗿𝗼𝗺?  
→ Stanford CS336 / CS236 / CS229 (Google it)  
→ MIT 6.S191, Karpathy’s Zero to Hero: [**https://lnkd.in/dT7vqqQ5**](https://lnkd.in/dT7vqqQ5)  
→ Google Kaggle GenAI sprint: [**https://lnkd.in/ga5X7tVJ**](https://lnkd.in/ga5X7tVJ)  
→ NVIDIA’s end-to-end LLM stack: [**https://lnkd.in/gCtDnhni**](https://lnkd.in/gCtDnhni)  
→ [**DeepLearning.AI**](http://deeplearning.ai/)’s short courses: [**https://lnkd.in/gAYmJqS6**](https://lnkd.in/gAYmJqS6)  
  
💥 𝗞𝗲𝗲𝗽 𝗶𝘁 𝗿𝗲𝗮𝗹:  
Don’t fall for “built in 5 min, dead in 10 min” demos.  
In prod, it’s about latency, cost, maintainability, guardrails.  
  
♻️ Let's repost to help more people on this journey 💚

