**Learning Roadmap for GenAI System Design**

**🔹 Layer 1 — Infra (Reliability & Security)**

**Goal:** Get FAANG-level confidence with AWS-based scalable and secure deployments.  
**Resources:**

* 📖 *Designing Data-Intensive Applications* (Martin Kleppmann) → gold standard for scaling patterns (queues, backpressure, circuit breakers).
* AWS Official Workshops:
  + [AWS Well-Architected Labs](https://wellarchitectedlabs.com/) (Reliability, Security pillars).
  + [ECS Workshop](https://ecsworkshop.com/) & [EventBridge Workshop](https://eventbridgeworkshop.com/).
* YouTube: *re:Invent talks* — “Advanced VPC Design”, “Scaling with SQS/EventBridge”, “Modern App Security on AWS”.
* System Design Fundamentals:
  + [Grokking the System Design Interview (Educative)](https://www.educative.io/courses/grokking-modern-system-design-interview-for-engineers-managers)
  + Alex Xu’s *System Design Interview Vol 1 & 2* (esp. caching, load balancers, autoscaling).

**🔹 Layer 2 — RAG (Knowledge Access & Grounding)**

**Goal:** Understand retrieval quality, embeddings, and hybrid strategies deeply.  
**Resources:**

* 📖 *The RAG Book* by Pinecone (free PDF).
* 📖 *Haystack RAG Playbook* (Deepset) → practical retrieval + hybrid pipelines.
* Papers with Code → *Self-ask*, *HyDE*, *ColBERTv2*.
* Tutorials:
  + LangChain docs (v0.3 modular retrieval).
  + LlamaIndex “Context Engineering” notebooks.
  + Pinecone / Weaviate / Milvus blog posts on hybrid retrieval & sharding.
* Benchmarks:
  + HuggingFace beir dataset for RAG eval.
  + DeepEval / TruLens frameworks.

**🔹 Layer 3 — Orchestration (Reason & Act)**

**Goal:** Master frameworks like LangGraph & AutoGen for DAG-based and multi-agent workflows.  
**Resources:**

* [LangGraph Docs](https://www.langchain.com/langgraph) → DAG execution, memory, guards.
* [Microsoft AutoGen](https://microsoft.github.io/autogen/) → multi-agent orchestration patterns.
* CrewAI framework (open-source alternative to AutoGen).
* 📖 “Agents & Tool Use in LLMs” survey (arXiv 2024).
* YouTube: *LangGraph office hours / tutorials* (LangChain team uploads weekly).
* Hands-on: build small DAGs (retriever → reasoner → tool → summarizer).

**🔹 Layer 4 — Eval & Guardrails (Trust & Safety)**

**Goal:** Measure & enforce reliability with metrics, eval loops, and guardrails.  
**Resources:**

* [TruLens](https://www.trulens.org/) → trace-based eval for RAG.
* [DeepEval](https://docs.confident-ai.com/) → win-rate, factuality, safety evals.
* [Guardrails AI](https://www.guardrailsai.com/) → structured outputs & policies.
* Papers: *Holistic Evaluation of Language Models* (Anthropic), *Constitutional AI*.
* AWS re:Invent “AI Governance and Guardrails” (2023).
* OpenAI eval blog posts on preference modeling.

**🔹 50+ GenAI System Design Patterns**

You won’t find all 50 in a single source. You’ll need to **synthesize across:**

1. **System design books** (for infra patterns → autoscaling, circuit breakers, backpressure).
2. **Retrieval frameworks** (for RAG patterns → hybrid, dual-stage).
3. **Agent frameworks** (for orchestration → DAGs, tool budget caps).
4. **Eval/Guardrail frameworks** (TruLens, Guardrails, DeepEval).
5. **Industry blogs** (OpenAI, Anthropic, Cohere, Pinecone, Weaviate).

👉 Best curated sources for *patterns specifically*:

* *AWS Builders Library* (patterns like backpressure, retries, failover).
* Pinecone blog (“RAG design patterns” series).
* LangChain blog (LangGraph orchestration patterns).
* Anthropic’s “Red-teaming LLMs” articles.
* Google’s *Site Reliability Engineering* book (patterns like bulkhead, rate limiting).

**🎯 Suggested Sequence (How to Learn)**

1. **Base Infra** (2–3 weeks): AWS workshops + Kleppmann → practice deploying RAG app on ECS/Fargate.
2. **RAG Layer** (2–3 weeks): Pinecone/Weaviate tutorials + BEIR benchmarks → try hybrid retrieval & rerankers.
3. **Orchestration** (2 weeks): LangGraph DAGs + AutoGen multi-agent → practice building workflows.
4. **Eval/Guardrails** (2 weeks): TruLens + DeepEval → setup dashboards, red-team tests.
5. **Patterns Deep Dive**: map each of your **50 patterns** into real case studies → e.g., Hybrid retrieval → Pinecone blog, Canary deploy → AWS Builders Library.

✅ By the end, you’ll not only *know* the GenAI system design layers & patterns, but also have **working AWS-based prototypes + evaluation dashboards**, which is exactly what FAANG-grade interviewers look for.