Skills Series

@das-purnendu

AWS SKILLS THAT GET GENAI ENGINEERS HIRED

Master AWS.
Build with GenAI.
Ship real impact.

Cloud + GenAI = Your unfair advantage.
Build, deploy and scale intelligent AI systems on AWS.

1. UNDERSTAND
Use GenAI to solve real-world problems

2. BUILD
Design RAG/LLM apps with the right data

3. ORCHESTRATE
Use AWS services to orchestrate intelligent workflows

4. DEPLOY
Deploy securely and scale with AWS

5. OPTIMIZE
Monitor, evaluate and continuously improve

POWERED BY AWS + GENAI

- Amazon Bedrock
- AWS Lambda
- Amazon S3
- Amazon OpenSearch Service
- Amazon API Gateway
- AWS Step Functions
- Amazon CloudWatch

FOR:

- GenAI Engineers
- AWS Learners
- Data Scientists
- Developers
- Cloud Enthusiasts
- Job Seekers

The future is GenAI.
The foundation is AWS.
Your skills make it real.

Secure.
Scalable.
Cost-effective.
Production-ready.

Real-world insights. Practical frameworks. Job-ready skills.

@das-purnendu

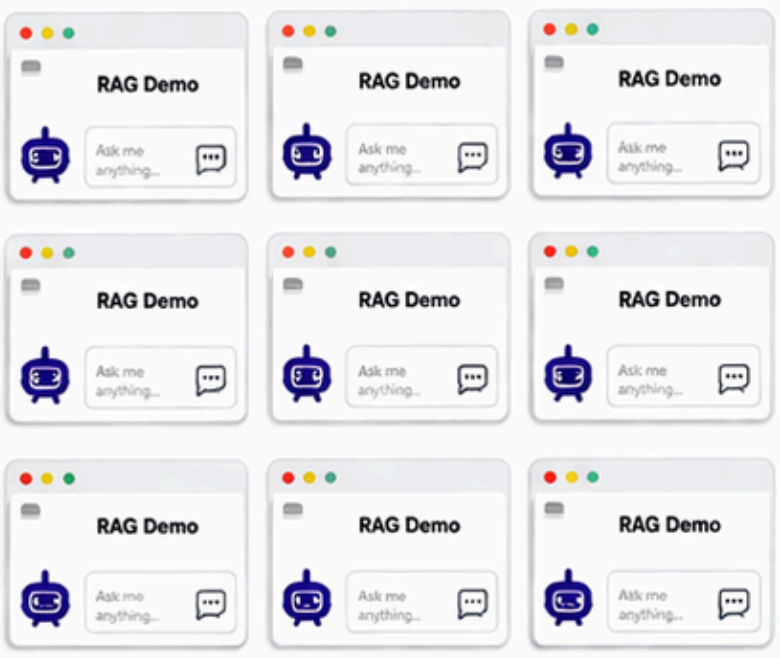
REALITY CHECK

EVERY CANDIDATE BUILDS RAG.

It's not about building a demo. It's about building what survives.

- Everyone has a demo.
- Few can run it in production.
- That's where hiring decisions happen.

WHAT 90% BUILD



Looks good. Doesn't scale.

WHAT GETS YOU HIRED

Reliable. Secure. Scalable. That's production.

Companies don't hire for demos. They hire for impact.

@das-purnendu

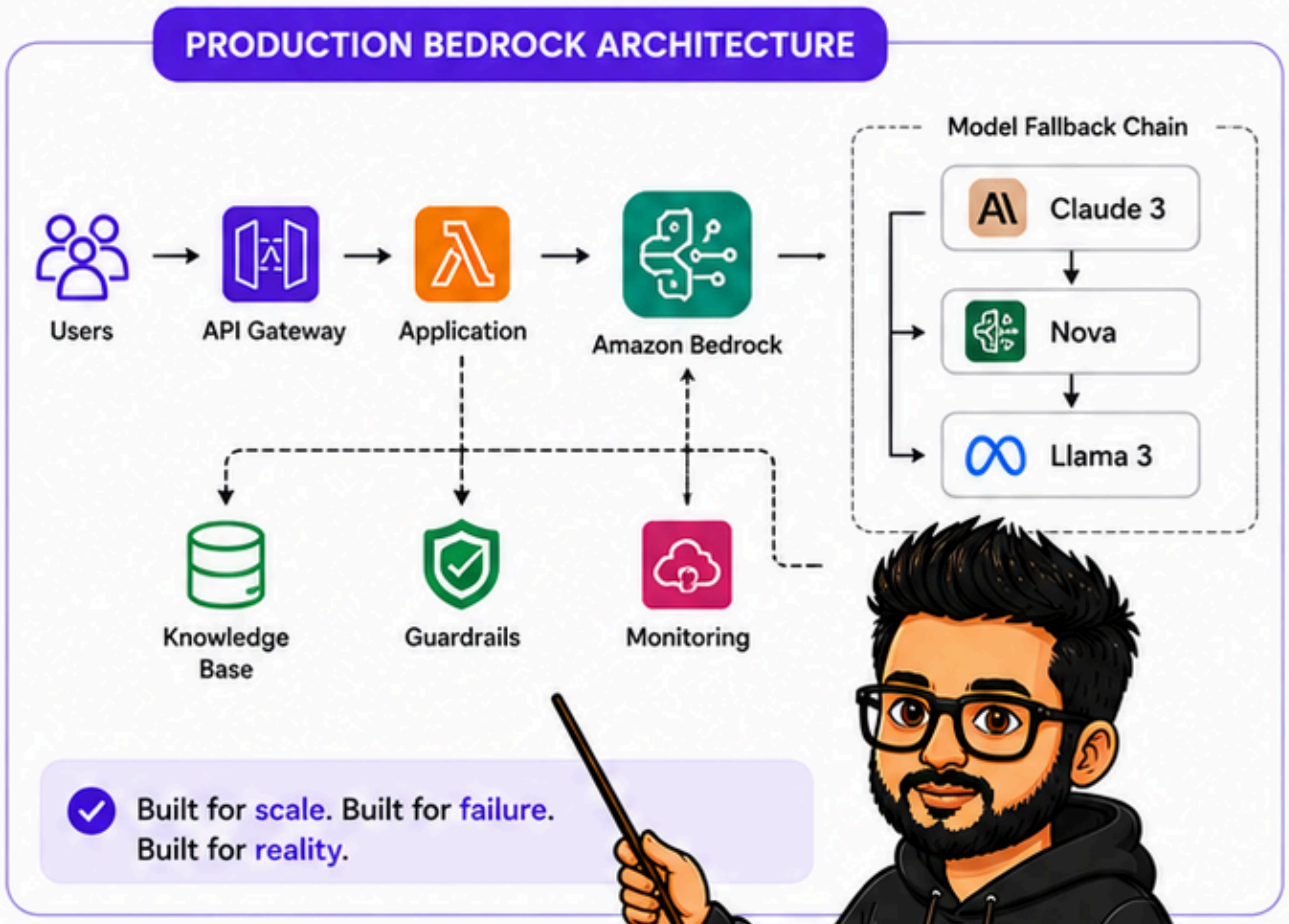
SKILL 1

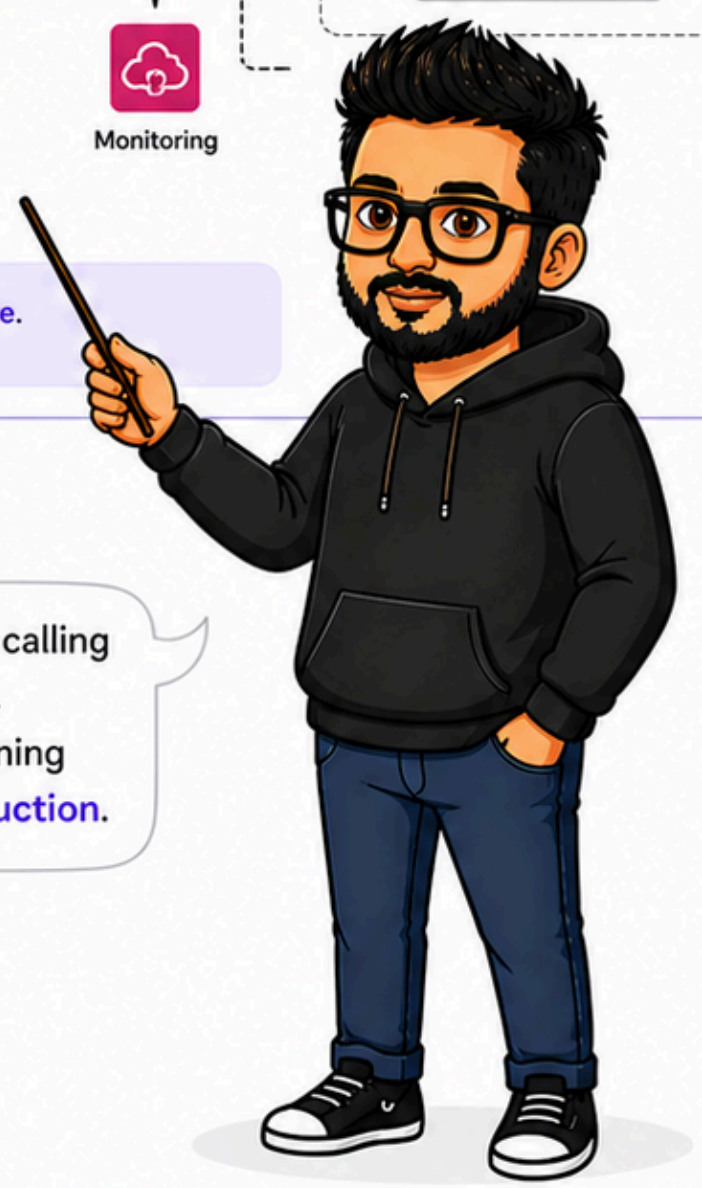
BEDROCK IN PRODUCTION

The notebook → production gap is where **80%** of GenAI candidates fall apart.

WHAT TO ACTUALLY KNOW:

- Provisioned Throughput vs On-Demand — and the **math** for when PT pays off
- Cross-region inference profiles for throughput + **failover**
- Bedrock **Guardrails**: PII redaction, denied topics, contextual grounding checks
- Knowledge Bases vs roll-your-own RAG — and the honest **tradeoffs**
- Model **fallback** chains (Claude → Nova → Llama) when your primary throttles





INTERVIEW QUESTION YOU'LL GET:

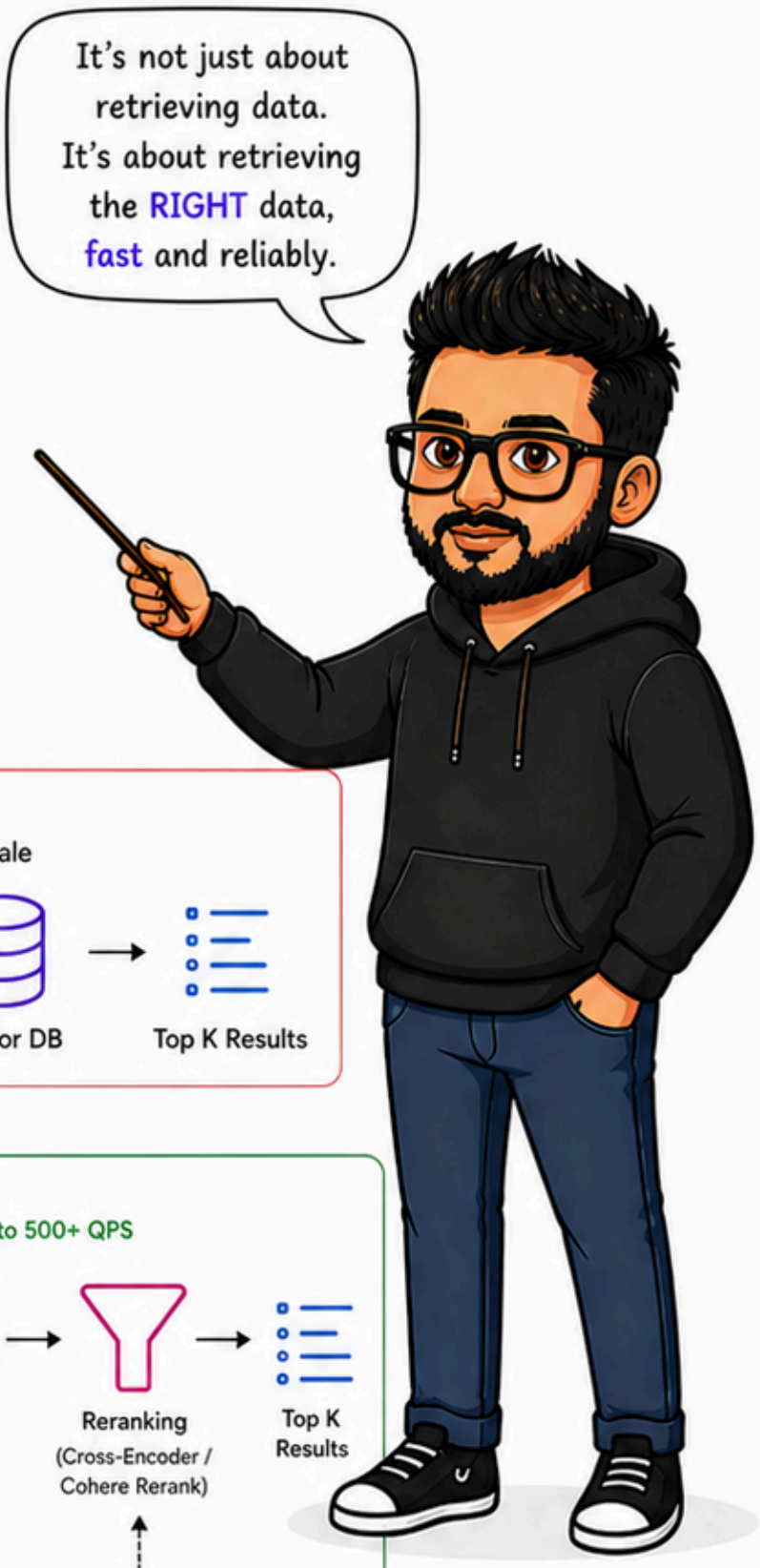
"Your Bedrock-backed chatbot just got **rate-limited** at 9 AM Monday. What's your architecture so this never happens again?"

It's not about calling **InvokeModel**.
It's about running LLMs in **production**.

Production mindset > Demo mindset.

SKILL 2

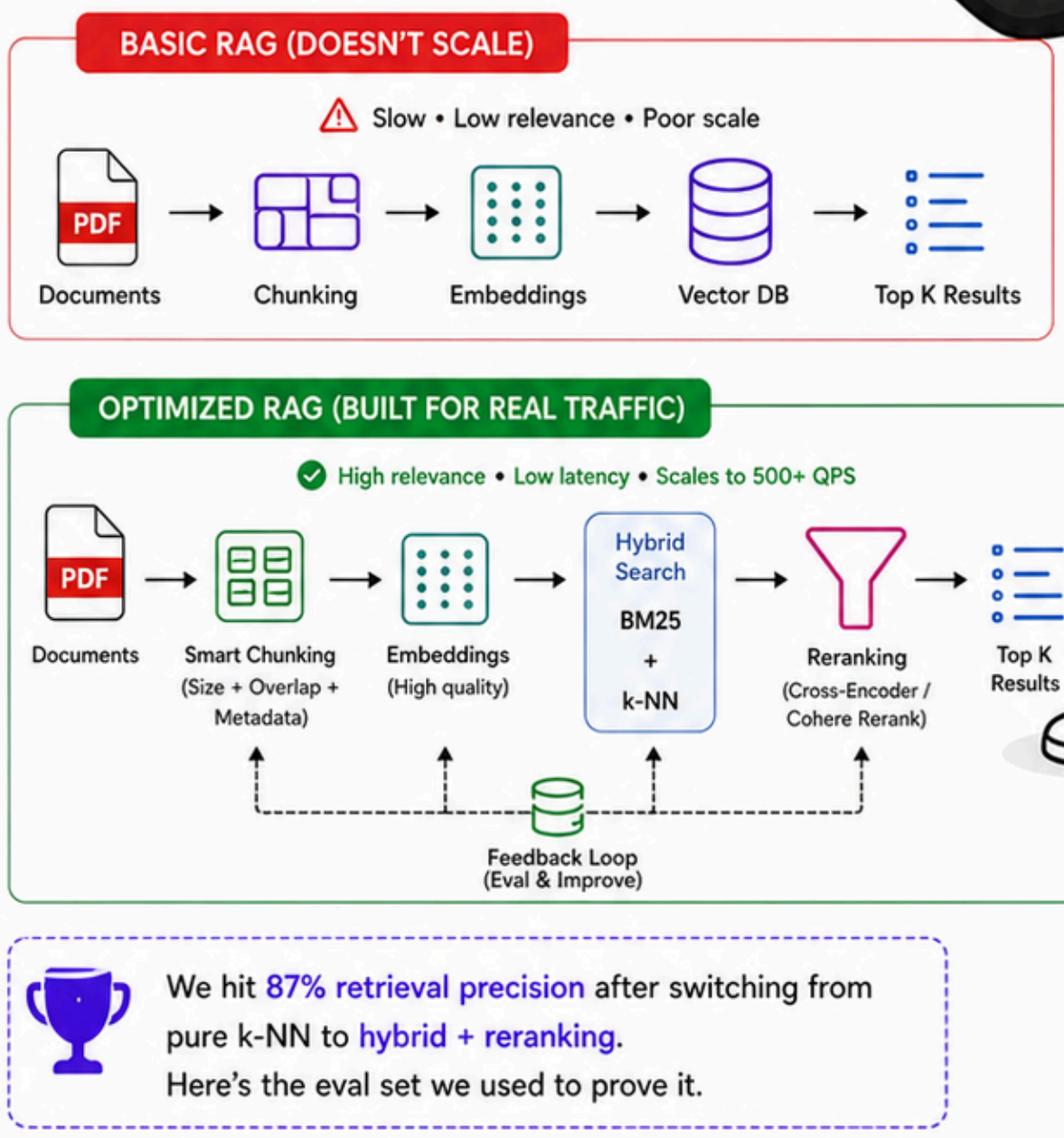
RAG THAT HANDLES REAL TRAFFIC



Anyone can chunk a PDF.
Few can run retrieval at **500 QPS** without latency falling apart.

WHAT TO KNOW:

- Hybrid search (BM25 + k-NN) with proper **score normalization**
- **Reranking** layer (Cohere Rerank, cross-encoders) and when it's worth the latency
- Chunk size, overlap & **metadata** filtering — with reasons, not vibes
- Embedding pipelines & **re-embedding** strategy when you swap models
- Measure what matters: retrieval **precision**, **recall@k**, **MRR** — not just "it works"



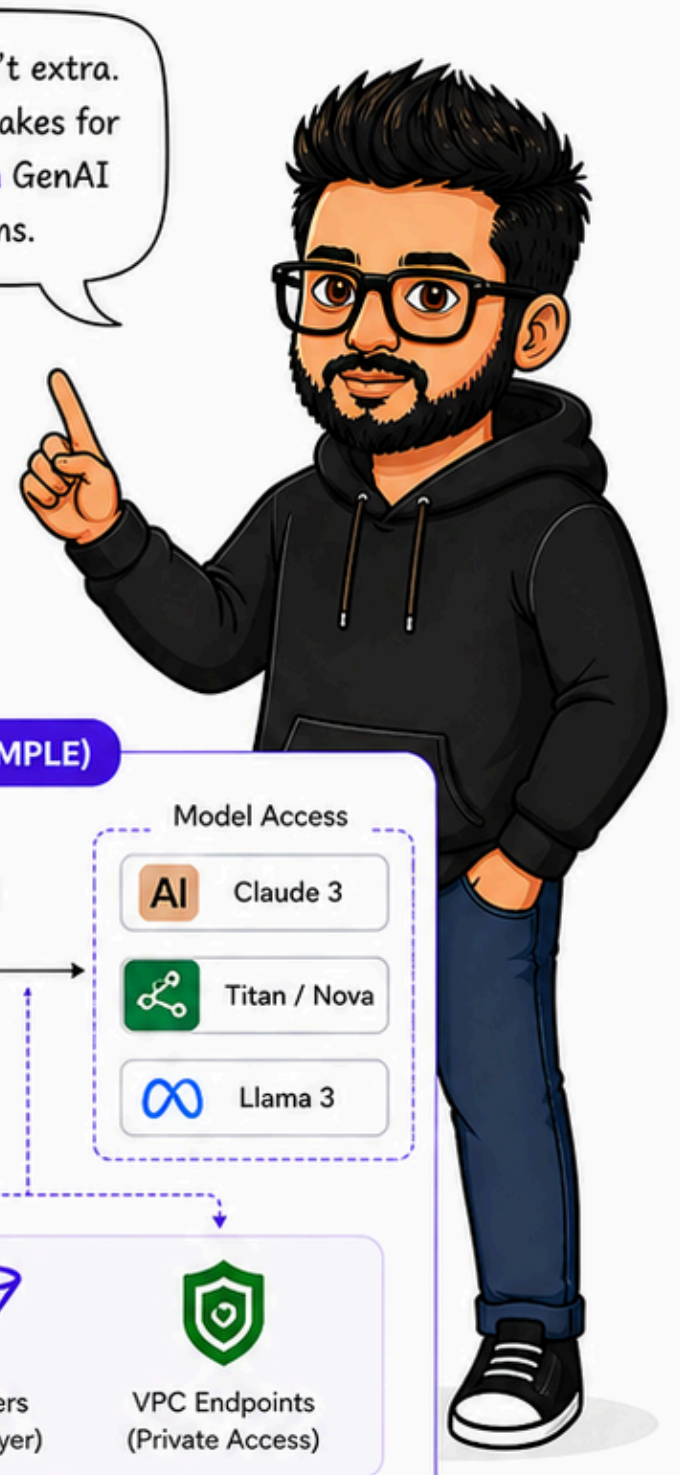
We hit **87% retrieval precision** after switching from pure k-NN to **hybrid + reranking**.
Here's the eval set we used to prove it.

Great RAG isn't luck. It's **architecture + evaluation + iteration.**

SKILL 3

SECURITY = JOB OFFERS 💰

Most candidates ignore security. The ones who don't **get hired** in regulated industries.



WHAT TO KNOW:

IAM least privilege
Design roles that **can't be** abused.

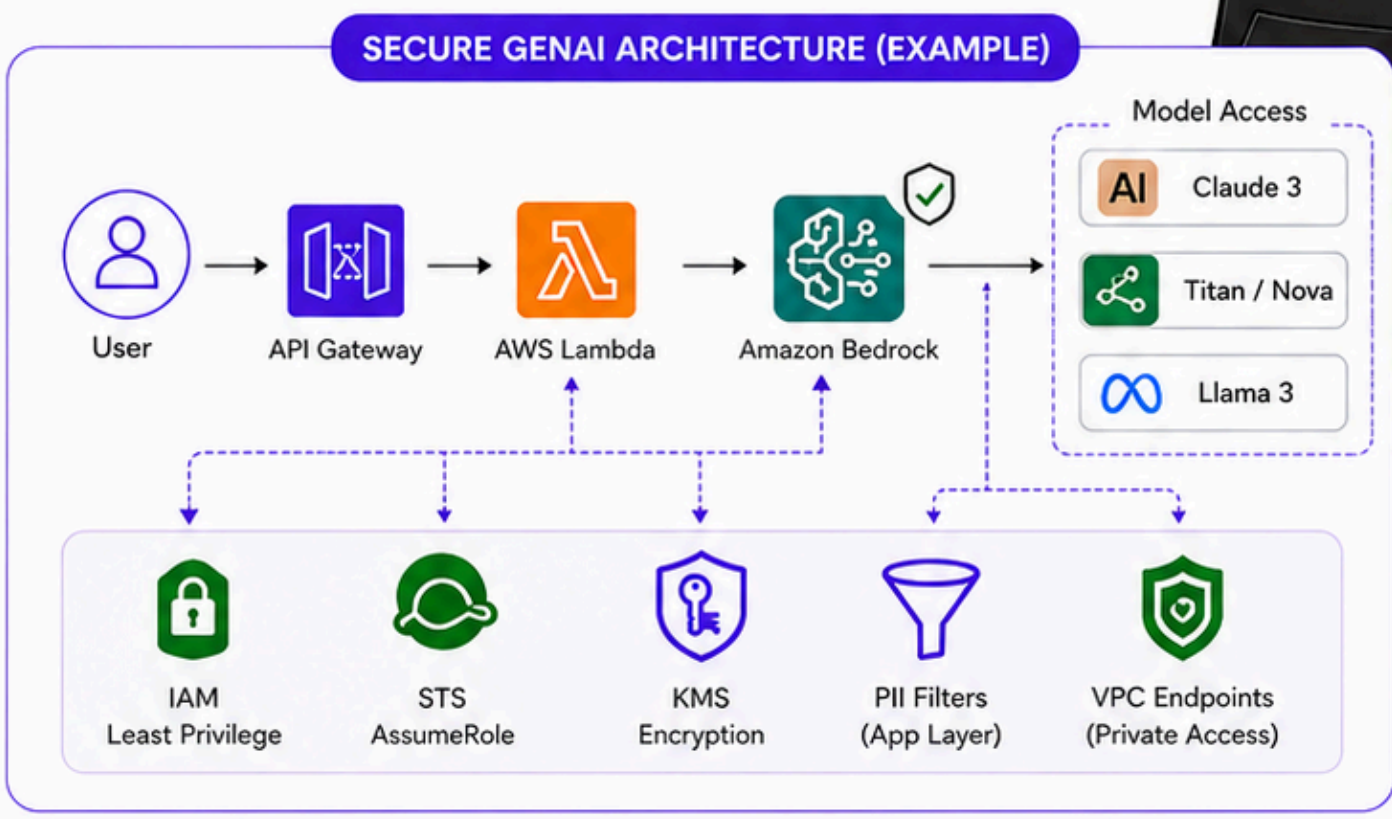
Cross-account access
Use **sts:AssumeRole** for Bedrock & other resources.

VPC endpoints for Bedrock
Keep data **private**.
Keep it in your **VPC**.

PII protection
KMS encryption for prompts & responses in CloudWatch.

Defense in depth
Bedrock **Guardrails** + custom PII filters in your app layer.

Prompt injection defense
"Ignore previous instructions" is the **SQL injection** of 2026.



DON'T DO THIS

Logging raw prompts & responses with PII in CloudWatch.

DO THIS INSTEAD

Redact PII, encrypt logs, and store only what you need to debug.

INTERVIEW QUESTION YOU'LL GET:

"How do you stop a user from exfiltrating your system prompt or other users' data through your RAG app?"

Senior answers cover: input validation, output filtering, retrieval-time access control, and monitoring — **not just Guardrails.**

Security isn't optional. It's the reason **enterprises hire you.**

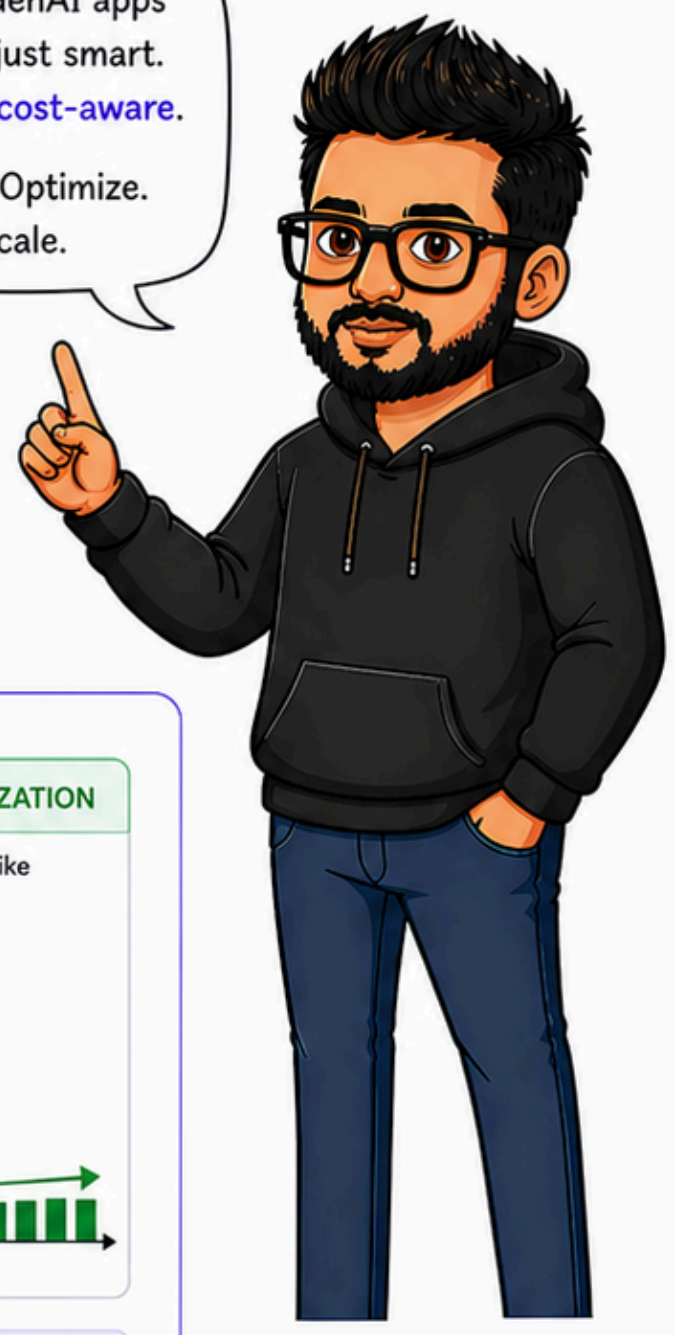
@das-purnendu

SKILL 4

COST CONTROL = SURVIVAL

Great GenAI apps aren't just smart. They're cost-aware.

Track. Optimize. Scale.



Your Bedrock bill can go from $200 to $20,000 in a weekend. Engineers who understand this get promoted; engineers who don't get fired.

WHAT TO KNOW:

- Token-level cost tracking per user, feature, tenant (not just monthly totals)
- Prompt caching: Up to 90% input cost reduction when used right
- Semantic caching for repeated queries (Redis + embedding similarity)
- Model routing: Cheap models for classification, expensive ones for generation
- Streaming responses: Better perceived latency + lower compute waste

THE COST DIFFERENCE

Same user experience. Same quality. 10X lower cost.

HOW TO CUT COST WITHOUT LOSING QUALITY

1. **Measure** — Track tokens & cost per user / feature / tenant
2. **Cache Smartly** — Use prompt caching + semantic caching aggressively
3. **Route Intelligently** — Use the right model for the right task (not one-size-fits-all)
4. **Stream Responses** — Improve UX + reduce wasted compute (tokens you don't need)
5. **Optimize Continuously** — Review dashboards, set budgets, iterate constantly

PORTFOLIO LINE THAT GETS RECRUITER DMs:

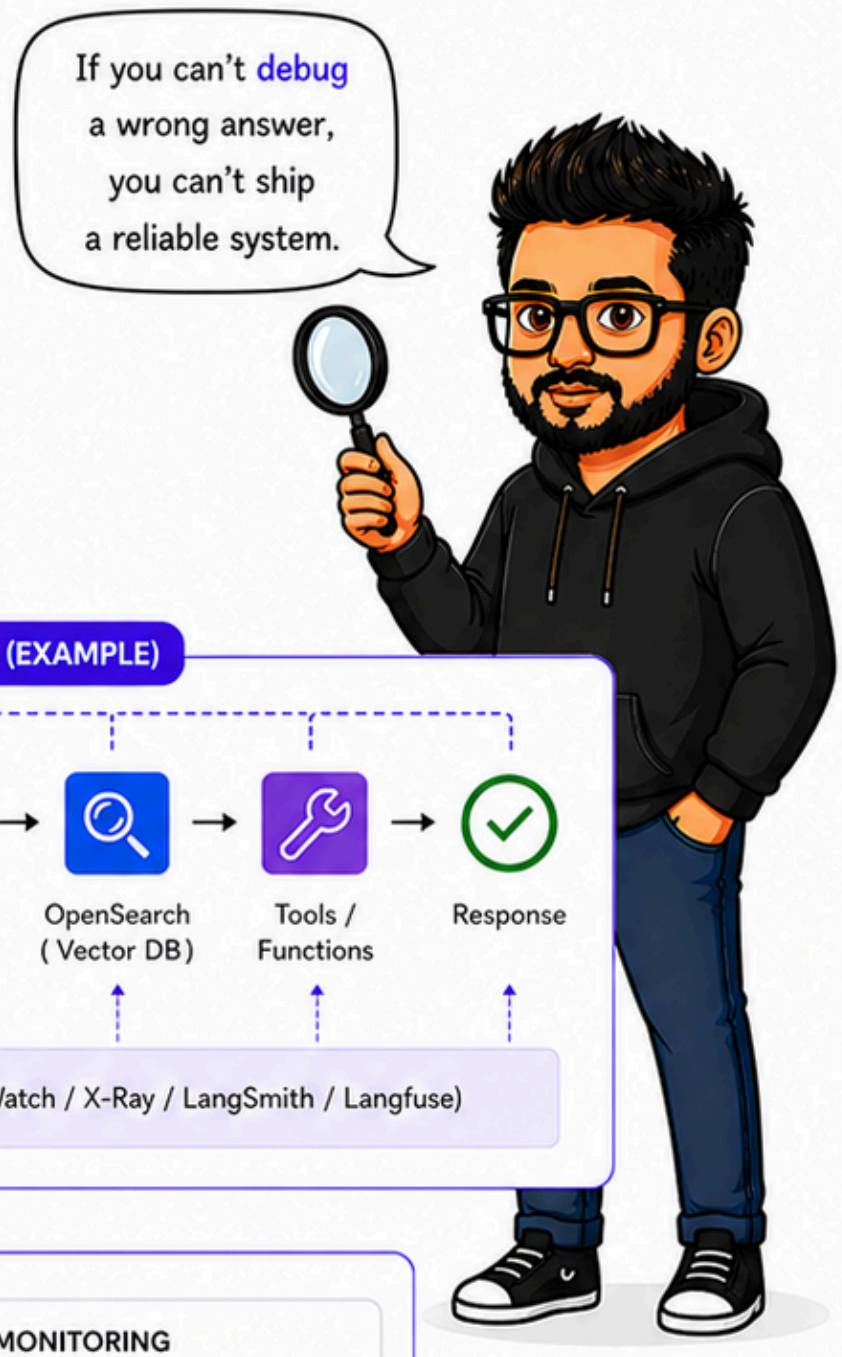
"Cut our RAG inference cost from $0.18 to $0.04 per query through prompt caching + model routing — same eval scores."

Cost control isn't about spending less. It's about scaling more.

@das-purnendu

SKILL 5

OBSERVABILITY + EVALS = SENIOR LEVEL 🚀



Dashboards won't tell you **why** the model hallucinated.
Observability + evals will.

WHAT TO KNOW:

End-to-end tracing across the agent loop:
API Gateway → Lambda → Bedrock → OpenSearch → tools (X-Ray / LangSmith)

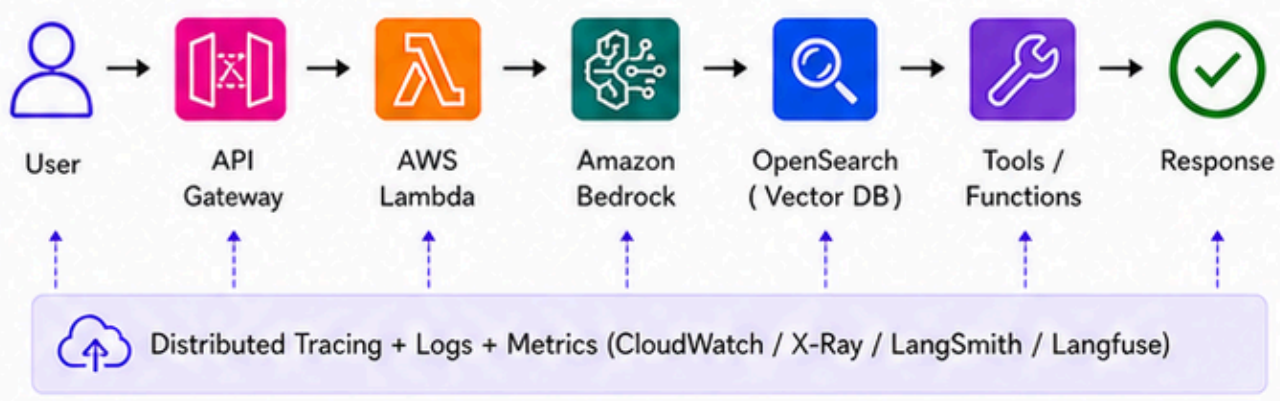
Offline eval sets with ground truth.
CI/CD that **blocks regressions**.

Online metrics that matter:
Groundedness, Answer Relevance, Retrieval Recall@k, Tool Success Rate

Drift detection: Embeddings drift, retrieval quality silently dies.

Log everything (the right way):
Prompts, retrieved chunks, tool calls, outputs — with **PII redaction**.

AGENT REQUEST FLOW (EXAMPLE)



EVALUATION THAT ACTUALLY WORKS

OFFLINE EVAL EXAMPLE

- ☑ Question
- ☑ Ground Truth
- ☑ Retrieved Context
- ☑ Model Answer
- ☑ Score (0–1)

ONLINE MONITORING

Groundedness	> 0.8
Answer Relevance	> 0.75
Retrieval Recall@k	> 0.7
Tool Success Rate	> 0.95
Latency (p95)	< 2.5s

INTERVIEW QUESTION YOU'LL GET:

"A user complains your agent gave a wrong answer yesterday at 3:47 PM. Walk me through how you'd reproduce and fix it."

- ✅ Find the request (logs + trace)
- ✅ Replay with same inputs
- ✅ Inspect retrieved chunks
- ✅ Check model + prompts version
- ✅ Identify root cause
- ✅ Fix + add eval to prevent reoccurrence

You can't improve what you can't measure.
Observability turns LLM apps from demos into dependable systems.

Debug faster. Ship safer.
Earn trust.

BONUS SLIDE 🎁

BEDROCK IN PRODUCTION: RESOURCES TO LEVEL UP

Keep learning.
Keep building.
Keep shipping. 🚀

Use the right resources.
Learn from real-world builders.

TOP RESOURCES I RECOMMEND

AWS Bedrock Docs
Official docs, best practices, and real examples
docs.aws.amazon.com/bedrock

Bedrock Workshops
Hands-on labs to build production GenAI apps
aws.amazon.com/workshops

AWS Architecture Center
Reference architectures for generative AI
aws.amazon.com/architecture

Awesome Bedrock (GitHub)
Curated list of projects, tools & samples
github.com/awesome-bedrock

YouTube: AWS Developers
Deep dives, demos and expert talks
youtube.com/@AWSDevelopers

BEDROCK PRODUCTION CHECKLIST

- [x] Designed RAG for your use case
- [x] Tested with real data & edge cases
- [x] Secured access, data & prompts
- [x] Added guardrails & content filters
- [x] Implemented cost controls
- [x] Monitoring, alerts & evals in place
- [x] Documented runbooks & SOPs
- [x] Continuous improvement loop

You don't need perfect.
You need shippable + improvable.

IF THIS WAS HELPFUL, PLEASE:

Like If you learned something new

Comment What will you build with Bedrock?

Share With someone who needs this

Save For your next project

Follow @das-purnendu
For more real-world AI content

Build. Ship. Iterate. That's how you win in the GenAI era.

Keep Building!

@das-purnendu

THANK YOU FOR READING!

I create content that helps you build **skills**, build **projects** and build your **future** in **GenAI**.

Keep learning.
Keep building.
Keep **shipping.** 🚀

IF YOU FOUND THIS HELPFUL, PLEASE:

LIKE
It helps more people see this

COMMENT
Share your thoughts or questions

SHARE
With someone who needs this

SAVE
For your next project or interview

FOLLOW
@das-purnendu
For more real-world GenAI insights, frameworks & career guidance.

YOUR JOURNEY TO AI EXCELLENCE STARTS NOW.

COMMENT "AI ROADMAP"
I'll send you a detailed roadmap to become a **GenAI Engineer**.

Let's build the future together! ♡