WHY Rabbit MQ?  
Handles async queries, decouples frontend & processing, supports multiple users.

Why Redis?  
Caching repeated questions, reduces RAG processing, improves latency.  
  
How does live streaming work?  
WebSocket keeps connection open; workers push results back as soon as available  
  
How to scale workers?  
Increase replicas in Kubernetes; optionally use **KEDA** to auto-scale based on queue length.  
  
RAG engine internals?  
Load PDF, embed content, retrieve relevant sections, generate answer using LLM.  
  
How to deploy on AWS?  
Docker images → ECR, Kubernetes manifests → EKS, use RDS/S3 in production.  
  
Why use Docker Compose locally?  
Quick local testing of Flask, RabbitMQ, Redis, and workers.  
  
Multiple LLM support?

Use smaller LLM for simple queries, GPT-3.5 for normal, GPT-4 for complex.

**PROJECT-DEFINITION**

"rag\_results\_channel" is the exact name of the Redis Pub/Sub channel that the server listens to.  
Any worker that publishes results must publish to the same channel: