# Cloud Architecture Test - Evaluation Scoring Grid

## Scoring Overview

Each deliverable is scored on a 1–5 scale where:

* 5 = Exceptional — Exceeds expectations, shows deep expertise
* 4 = Strong — Meets all requirements with good understanding
* 3 = Adequate — Basic requirements met, some gaps or unclear areas
* 2 = Weak — Missing key components, fundamental misunderstandings
* 1 = Poor — Major gaps, incorrect approach, unusable solution

## 1. RAG Architecture Diagram Evaluation

|  |  |  |
| --- | --- | --- |
| Score | Criteria | What This Looks Like |
| 5 | Complete end-to-end flow with optimization details | * Shows document ingestion: chunking, embedding, vector storage, retrieval, context assembly, LLM generation, response * Includes preprocessing steps (OCR, text extraction) * Shows feedback loops for improving retrieval * Indicates caching layers and performance optimizations * Clear data lineage and error handling paths |
| 4 | Solid RAG pipeline with most key components | * Shows main RAG flow: ingest, embed, store, retrieve, generate * Includes vector database and LLM components * Shows user interface connection * Some optimization considerations mentioned * Minor gaps in preprocessing or error handling |
| 3 | Basic RAG flow understanding | * Shows documents, embeddings, search, response flow * Includes vector database and LLM * Missing some preprocessing steps * Limited optimization considerations * Diagram is functional but not comprehensive |
| 2 | Incomplete understanding of RAG | * Shows some components but flow is unclear * Missing key steps like embedding or chunking * Confuses traditional search with vector search * Limited connection between components |
| 1 | Fundamental misunderstanding | * No clear RAG pipeline shown * Missing vector database entirely * Treats as traditional keyword search * Major architectural components missing |

## 2. Technology Stack Evaluation

|  |  |  |
| --- | --- | --- |
| Score | Criteria | What This Looks Like |
| 5 | Optimal choices with detailed justification | * Specific vector DB choice with reasoning * LLM strategy with cost/performance tradeoffs * Embedding model selection rationale * Cloud services mapped to specific functions * Alternative options discussed with pros/cons |
| 4 | Good technology choices with some reasoning | * Appropriate vector database selection * Reasonable LLM choice for use case * Cloud services align with requirements * Some justification for major choices * Considers cost vs performance |
| 3 | Adequate technology selection | * Names appropriate technologies * Basic understanding of vector databases * Standard cloud services mentioned * Limited justification for choices * May miss some optimization opportunities |
| 2 | Poor technology alignment | * Inappropriate choices for scale/budget * Limited understanding of vector databases * Generic cloud services without specificity * No cost considerations * Technology doesn't match requirements |
| 1 | Wrong or missing technology choices | * No vector database mentioned * Inappropriate or outdated technologies * No consideration of RAG-specific needs * Major technology gaps |

## 3. Security Architecture Diagram Evaluation

|  |  |  |
| --- | --- | --- |
| Score | Criteria | What This Looks Like |
| 5 | Comprehensive security with enterprise considerations | * Multi-layer security diagram (network, application, data) * Shows SSO (single sign on) integration and role-based access * Document-level permissions mapping * Audit logging throughout pipeline * Data encryption at rest and in transit * Zero-trust principles applied |
| 4 | Strong security architecture | * Shows authentication and authorization layers * Includes encryption and access controls * Document permissions considered * Audit logging included * Minor gaps in defense-in-depth |
| 3 | Basic security measures | * Standard authentication (SSO or similar) * Basic access controls * Some encryption mentioned * Limited document-level security * Basic audit considerations |
| 2 | Minimal security awareness | * Only basic authentication * Limited access controls * Missing encryption details * No document-level permissions * Minimal audit capabilities |
| 1 | Poor security understanding | * No clear security architecture * Missing authentication details * No access controls shown * Security treated as afterthought |

## 4. Scaling Strategy Evaluation

|  |  |  |
| --- | --- | --- |
| Score | Criteria | What This Looks Like |
| 5 | Comprehensive scaling with specific metrics and triggers | * Clear scaling triggers (e.g., “when vector search latency > 2s, scale horizontally”) * Database sharding strategy for growing corpus * Auto-scaling configurations for compute * Caching strategies (query cache, embedding cache) * Performance monitoring and alerting * Capacity planning with growth projections * Cost optimization during scaling |
| 4 | Good scaling approach with key considerations | * Horizontal scaling strategy defined * Database scaling approach clear * Some performance optimization * Monitoring considerations * Growth trajectory understood |
| 3 | Basic scaling understanding | * General scaling approach mentioned * Some auto-scaling concepts * Limited optimization strategies * Basic growth considerations |
| 2 | Limited scaling awareness | * Vague scaling mentions * No specific strategies * Limited understanding of bottlenecks * Poor growth planning |
| 1 | No scaling strategy | * No scaling considerations * Static architecture * No growth planning * Major scalability oversights |

## 5. Cost Strategy Evaluation

|  |  |  |
| --- | --- | --- |
| Score | Criteria | What This Looks Like |
| 5 | Detailed cost optimization with specific tactics | * Breakdown of major cost centers (LLM API, vector DB, compute, storage) * Specific cost reduction strategies (e.g., “use smaller embedding models for 80% of queries, GPT‑4 only for complex questions”) * Caching strategy to reduce API calls * Open-source alternatives evaluated * Development vs production cost considerations * ROI projections and cost monitoring plan |
| 4 | Good cost awareness with optimization strategies | * Major cost components identified * Some optimization strategies (caching, model selection) * Considers development cost constraints * Basic cost monitoring approach * Balances cost vs performance |
| 3 | Basic cost considerations | * Aware of major costs (API calls, infrastructure) * Some cost-saving mentions * Limited optimization strategies * Basic budgeting awareness |
| 2 | Minimal cost awareness | * Vague cost mentions * No specific optimization strategies * Poor understanding of cost drivers * Limited budget considerations |
| 1 | No cost strategy | * No cost considerations * Expensive choices without justification * No optimization awareness * Budget constraints ignored |

## 6. Implementation Phases Evaluation

|  |  |  |
| --- | --- | --- |
| Score | Criteria | What This Looks Like |
| 5 | Strategic phasing with clear dependencies and risk mitigation | * 3–4 logical phases with clear success criteria * Dependencies between phases identified * MVP approach for early value delivery * Risk mitigation built into phasing * Parallel development opportunities identified * User feedback integration points * Resource allocation considerations |
| 4 | Well-structured phases with good prioritization | * Clear phase breakdown with priorities * MVP approach understood * Dependencies mostly identified * Good feature prioritization * Some risk considerations |
| 3 | Basic phasing with adequate prioritization | * Phases identified but may lack detail * Basic MVP concept * Some prioritization logic * Limited dependency mapping |
| 2 | Poor phasing strategy | * Unclear phases or priorities * No MVP thinking * Dependencies not considered * Illogical feature sequencing |
| 1 | No implementation strategy | * • No clear phases * • All features at once approach * • No prioritization * • Poor project planning |

## Score Summary (Copy Totals Here)

|  |  |
| --- | --- |
| Section | Score |
| 1. RAG Architecture Diagram |  |
| 2. Technology Stack |  |
| 3. Security Architecture Diagram |  |
| 4. Scaling Strategy |  |
| 5. Cost Strategy |  |
| 6. Implementation Phases |  |
| Total |  |