

RAG Implementation with Amazon Kendra and Bedrock Cost Analysis Estimate Report

Service Overview

RAG Implementation with Amazon Kendra and Bedrock is a fully managed, serverless service that allows you to This project uses multiple AWS services.. This service follows a pay-as-you-go pricing model, making it cost-effective for various workloads.

Pricing Model

This cost analysis estimate is based on the following pricing model: - **ON DEMAND** pricing (pay-as-you-go) unless otherwise specified - Standard service configurations without reserved capacity or savings plans - No caching or optimization techniques applied

Assumptions

- Standard ON DEMAND pricing model for all services
- US East (N. Virginia) region for all resources
- Kendra Enterprise Edition for production-grade search capabilities
- Claude 3 Haiku model for cost-effective text generation
- Lambda functions with 512MB memory allocation
- API Gateway REST API for backend integration
- S3 Standard storage class for document storage
- No caching or optimization applied initially
- Average document size of 100KB
- Average query response time of 2 seconds

Limitations and Exclusions

- Data transfer costs between regions
- Custom model training costs for Bedrock
- Development and maintenance costs
- Third-party integration costs
- Backup and disaster recovery costs
- Monitoring and logging costs beyond basic CloudWatch
- CDK deployment infrastructure costs
- React frontend hosting costs (local development)

Cost Breakdown

Unit Pricing Details

Service	Resource Type	Unit	Price	Free Tier
Amazon Kendra	Enterprise Edition	hour	\$1.40	No free tier available for Kendra Enterprise Edition
Amazon Kendra	Document Scanning	document	\$0.000001	No free tier available for Kendra Enterprise Edition

Amazon Bedrock	Input Tokens	1,000 tokens	\$0.00025	No free tier for Bedrock foundation models
Amazon Bedrock	Output Tokens	1,000 tokens	\$0.00125	No free tier for Bedrock foundation models
AWS Lambda	Requests	request	\$0.0000002	First 12 months: 1M requests/month and 400,000 GB-seconds/month free
AWS Lambda	Compute	GB-second (Tier 1)	\$0.0000166667	First 12 months: 1M requests/month and 400,000 GB-seconds/month free
Amazon API Gateway	Rest Api Requests	request (first 333M requests)	\$0.0000035	First 12 months: 1M API calls/month free
Amazon S3	Standard Storage	GB-month (first 50TB)	\$0.023	First 12 months: 5GB Standard storage free

Cost Calculation

Service	Usage	Calculation	Monthly Cost
Amazon Kendra	Enterprise Edition with 1 index, processing 1000 documents, 500 queries per month (Runtime Hours: 730 hours/month (24/7 operation), Documents Processed: 1,000 documents)	$\$1.40/\text{hour} \times 730 \text{ hours}$ $+ \$0.000001 \times 1,000 \text{ documents} = \$1,022.00$ $+ \$0.001 = \$1,022.001$ $\approx \$1,022.00$	\$1,032.00
Amazon Bedrock	Claude 3 Haiku model processing 100K input tokens and 50K output tokens per month (Input Tokens: 100,000 tokens/month, Output Tokens: 50,000 tokens/month) 3 functions (ingest, query,	$\$0.00025/1K \times 100K$ $\text{input tokens} +$ $\$0.00125/1K \times 50K$ $\text{output tokens} = \$25.00$ $+ \$62.50 = \87.50	\$37.50

	sample-questions) with 1,000 requests/month,	$\$0.0000002 \times 1,000$	
AWS Lambda	512MB memory, 2s duration (Requests: 1,000 requests/month, Compute: 1,000 requests \times 2s \times 0.5GB = 1,000 GB-seconds/month)	requests + $\$0.0000166667 \times 1,000$ GB-seconds = $\$0.0002 + \$0.0167 = \$0.0169 \approx \0.02	\$0.03
Amazon API Gateway	REST API with 1,000 requests per month (Api Requests: 1,000 requests/month)	$\$0.0000035 \times 1,000$ requests = $\$0.0035 \approx \0.004	\$0.004
Amazon S3	Document storage with 1GB of documents (Storage: 1 GB-month)	$\$0.023 \times 1$ GB-month = $\$0.023$	\$0.023
Total	All services	Sum of all calculations	\$38.56/month

Free Tier

Free tier information by service: - **Amazon Kendra**: No free tier available for Kendra Enterprise Edition - **Amazon Bedrock**: No free tier for Bedrock foundation models - **AWS Lambda**: First 12 months: 1M requests/month and 400,000 GB-seconds/month free - **Amazon API Gateway**: First 12 months: 1M API calls/month free - **Amazon S3**: First 12 months: 5GB Standard storage free

Cost Scaling with Usage

The following table illustrates how cost estimates scale with different usage levels:

Service	Low Usage	Medium Usage	High Usage
Amazon Kendra	\$0/month	\$1/month	\$2/month
Amazon Bedrock	\$18/month	\$37/month	\$75/month
AWS Lambda	\$0/month	\$0/month	\$0/month
Amazon API Gateway	\$0/month	\$0/month	\$0/month
Amazon S3	\$0/month	\$0/month	\$0/month

Key Cost Factors

- **Amazon Kendra**: Enterprise Edition with 1 index, processing 1000 documents, 500 queries per month
- **Amazon Bedrock**: Claude 3 Haiku model processing 100K input tokens and 50K output tokens per month
- **AWS Lambda**: 3 functions (ingest, query, sample-questions) with 1,000 requests/month, 512MB memory, 2s duration
- **Amazon API Gateway**: REST API with 1,000 requests per month

- **Amazon S3:** Document storage with 1GB of documents

Projected Costs Over Time

The following projections show estimated monthly costs over a 12-month period based on different growth patterns:

Base monthly cost calculation:

Service	Monthly Cost			
Amazon Kendra	\$1.00			
Amazon Bedrock	\$37.50			
AWS Lambda	\$0.03			
Amazon API Gateway	\$0.00			
Amazon S3	\$0.02			
Total Monthly Cost \$38				
Growth Pattern	Month 1	Month 3	Month 6	Month 12
Steady	\$38/mo	\$38/mo	\$38/mo	\$38/mo
Moderate	\$38/mo	\$42/mo	\$49/mo	\$65/mo
Rapid	\$38/mo	\$46/mo	\$62/mo	\$110/mo

- Steady: No monthly growth (1.0x)
- Moderate: 5% monthly growth (1.05x)
- Rapid: 10% monthly growth (1.1x)

Detailed Cost Analysis

Pricing Model

ON DEMAND

Exclusions

- Data transfer costs between regions
- Custom model training costs for Bedrock
- Development and maintenance costs
- Third-party integration costs
- Backup and disaster recovery costs
- Monitoring and logging costs beyond basic CloudWatch
- CDK deployment infrastructure costs
- React frontend hosting costs (local development)

Recommendations

Immediate Actions

- Start with Kendra Developer Edition (\$1.125/hour) for development and testing to reduce initial costs
- Implement response caching in API Gateway to reduce Lambda invocations and Bedrock token usage
- Use S3 Intelligent Tiering for document storage to automatically optimize storage costs
- Monitor Kendra query patterns and optimize index configuration to reduce

unnecessary processing ##### Best Practices

- Implement CloudWatch monitoring to track usage patterns and optimize resource allocation
- Use Lambda provisioned concurrency only if consistent low latency is required
- Consider using Bedrock batch processing for bulk document analysis to reduce per-token costs
- Implement proper error handling to avoid unnecessary retries and associated costs
- Use CDK conditions to deploy different configurations for development vs production environments

Cost Optimization Recommendations

Immediate Actions

- Start with Kendra Developer Edition (\$1.125/hour) for development and testing to reduce initial costs
- Implement response caching in API Gateway to reduce Lambda invocations and Bedrock token usage
- Use S3 Intelligent Tiering for document storage to automatically optimize storage costs

Best Practices

- Implement CloudWatch monitoring to track usage patterns and optimize resource allocation
- Use Lambda provisioned concurrency only if consistent low latency is required
- Consider using Bedrock batch processing for bulk document analysis to reduce per-token costs

Conclusion

By following the recommendations in this report, you can optimize your RAG Implementation with Amazon Kendra and Bedrock costs while maintaining performance and reliability. Regular monitoring and adjustment of your usage patterns will help ensure cost efficiency as your workload evolves.