RAG Solution with Amazon Kendra and Bedrock Cost Analysis Estimate Report

Service Overview

RAG Solution 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 deployment
- Claude 3.5 Sonnet model for Bedrock responses
- Kendra Developer Edition for prototype/development
- Average query response size of 1,000 tokens
- Average document retrieval context of 2,000 tokens per query
- Lambda functions with 512 MB memory allocation
- API Gateway REST API usage (not HTTP API)
- No caching or optimization applied initially
- Serverless architecture with pay-per-use model

Limitations and Exclusions

- Data transfer costs between regions
- CloudWatch logging and monitoring costs
- S3 storage costs for document storage
- Development and maintenance costs
- Custom model training or fine-tuning costs
- Reserved capacity or savings plans discounts
- Enterprise support costs
- CDK deployment infrastructure costs

Cost Breakdown

Unit Pricing Details

Service	Resource Type	Unit	Price	Free Tier
Amazon Kendra	Developer Edition	hour	$1.125	No free tier available for Kendra
Amazon Kendra	Storage Capacity	hour per additional storage unit	$3.50	No free tier available for Kendra
Amazon Kendra	Query Capacity	hour per additional query unit	$3.50	No free tier available for Kendra
Amazon Bedrock - Claude 3.5 Sonnet	Input Tokens	1,000,000 tokens	$3.00	No free tier for Bedrock foundation models
Amazon Bedrock - Claude 3.5 Sonnet	Output Tokens	1,000,000 tokens	$15.00	No free tier for Bedrock foundation models
AWS Lambda	Requests	1,000,000 requests	$0.20	First 12 months: 1M requests/month and 400,000 GB-seconds/month free
AWS Lambda	Compute	GB-second	$0.0000166667	First 12 months: 1M requests/month and 400,000 GB-seconds/month free
Amazon API Gateway	Api Calls	1,000,000 requests (first 333M)	$3.50	First 12 months: 1M API calls/month free

Cost Calculation

Service	Usage	Calculation	Monthly Cost
Amazon Kendra	Developer Edition for document indexing and search (Base Hours: 720 hours/month (24/7 operation), Additional Storage: 0 units (included in base), Additional Queries: 0 units (included in base))	$1.125/hour × 720 hours = $810.00/month	$810.00/month
Amazon Bedrock - Claude 3.5 Sonnet	AI response generation with context-aware responses (Input Tokens: 3,000,000 tokens/month (3K per query × 1,000 queries), Output Tokens: 1,000,000 tokens/month (1K per query × 1,000 queries))	($3.00/1M × 3M input) + ($15.00/1M × 1M output) = $9.00 + $15.00 = $24.00/month for 1,000 queries	$45.00/month (1,000 queries)

Service	Usage	Calculation	Monthly Cost
AWS Lambda	Backend API processing with 512MB memory allocation (Requests: 1,000 requests/month, Compute: 1,000 requests × 2s × 0.5GB = 1,000 GB-seconds)	($0.20/1M × 0.001M requests) + ($0.0000166667 × 1,000 GB-seconds) = $0.0002 + $0.0167 = $0.017/month (within free tier)	$0.83/month (1,000 requests)
Amazon API Gateway	REST API for frontend-backend communication (Api Calls: 1,000 requests/month)	$3.50/1M × 0.001M requests = $0.0035/month (within free tier)	$3.50/month (1,000 requests)
Total	**All services**	**Sum of all calculations**	**$859.33/month**

Free Tier

Free tier information by service: - **Amazon Kendra**: No free tier available for Kendra - **Amazon Bedrock - Claude 3.5 Sonnet**: 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

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	$405/month	$810/month	$1620/month
Amazon Bedrock - Claude 3.5 Sonnet	$22/month	$45/month	$90/month
AWS Lambda	$0/month	$0/month	$1/month
Amazon API Gateway	$1/month	$3/month	$7/month

Key Cost Factors

- **Amazon Kendra**: Developer Edition for document indexing and search
- **Amazon Bedrock - Claude 3.5 Sonnet**: AI response generation with context-aware responses
- **AWS Lambda**: Backend API processing with 512MB memory allocation
- **Amazon API Gateway**: REST API for frontend-backend communication

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	$810.00
Amazon Bedrock - Claude 3.5 Sonnet	$45.00
AWS Lambda	$0.83
Amazon API Gateway	$3.50
Total Monthly Cost	**$859**

Growth Pattern	Month 1	Month 3	Month 6	Month 12
Steady	$859/mo	$859/mo	$859/mo	$859/mo
Moderate	$859/mo	$947/mo	$1096/mo	$1469/mo
Rapid	$859/mo	$1039/mo	$1383/mo	$2451/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
- CloudWatch logging and monitoring costs
- S3 storage costs for document storage
- Development and maintenance costs
- Custom model training or fine-tuning costs
- Reserved capacity or savings plans discounts
- Enterprise support costs
- CDK deployment infrastructure costs

Recommendations

Immediate Actions

- Start with Kendra Developer Edition for prototyping and development
- Monitor token usage patterns to optimize Bedrock costs
- Implement response caching to reduce repeated API calls
- Use Lambda ARM architecture for 20% cost savings on compute
- Consider API Gateway HTTP API instead of REST API for lower costs

Cost Optimization Recommendations

Immediate Actions

- Start with Kendra Developer Edition for prototyping and development
- Monitor token usage patterns to optimize Bedrock costs
- Implement response caching to reduce repeated API calls

Best Practices

- Regularly review costs with AWS Cost Explorer
- Consider reserved capacity for predictable workloads
- Implement automated scaling based on demand

Conclusion

By following the recommendations in this report, you can optimize your RAG Solution 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.