RAG Implementation Solution Cost Analysis Estimate Report

Service Overview

RAG Implementation Solution 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
- Claude 3.5 Haiku model for cost-effective AI responses
- Kendra Developer Edition for prototype/development use
- Lambda functions with 512MB memory allocation
- Moderate usage patterns for prototype implementation
- No reserved instances or savings plans applied
- Standard storage classes and configurations

Limitations and Exclusions

- Data transfer costs between regions
- CloudWatch monitoring and logging costs
- Development and maintenance costs
- Custom domain and SSL certificate costs
- Backup and disaster recovery costs
- Third-party integrations or tools
- Network infrastructure costs (VPC, NAT Gateway, etc.)
- Security services (WAF, Shield, etc.)

Cost Breakdown

Unit Pricing Details

Service	Resource Type	Unit	Price	Free Tier
Amazon Bedrock (Claude	Input Tokens	1,000 tokens	$0.0008	No free tier for Bedrock foundation

3.5 Haiku)				models
Amazon Bedrock (Claude 3.5 Haiku)	Output Tokens	1,000 tokens	$0.004	No free tier for Bedrock foundation models
Amazon Kendra (Developer Edition)	Base Capacity	hour	$1.125	No free tier for Kendra
AWS Lambda	Requests	request	$0.0000002	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 S3	Storage	GB per month	$0.023	First 12 months: 5GB storage, 20,000 GET requests, 2,000 PUT requests free
Amazon S3	Put Requests	1,000 requests	$0.005	First 12 months: 5GB storage, 20,000 GET requests, 2,000 PUT requests free
Amazon S3	Get Requests	1,000 requests	$0.0004	First 12 months: 5GB storage, 20,000 GET requests, 2,000 PUT requests free
Amazon DynamoDB	Read Requests	million requests	$0.125	Always free: 25GB storage, 25 read capacity units, 25 write capacity units
Amazon DynamoDB	Write Requests	million requests	$1.25	Always free: 25GB storage, 25 read capacity units, 25 write capacity units
				Always free:

Amazon DynamoDB	Storage	GB per month	$0.25	25GB storage, 25 read capacity units, 25 write capacity units
Amazon API Gateway	Http Requests	million requests	$1.00	First 12 months: 1M API calls per month free

Cost Calculation

Service	Usage	Calculation	Monthly Cost
Amazon Bedrock (Claude 3.5 Haiku)	Processing 100K input tokens and 50K output tokens per month (Input Tokens: 100,000 tokens, Output Tokens: 50,000 tokens)	$0.0008/1K × 100K input tokens + $0.004/1K × 50K output tokens = $0.08 + $0.20 = $0.28	$0.12
Amazon Kendra (Developer Edition)	24/7 operation for development and testing (Hours: 720 hours per month (24/7))	$1.125/hour × 720 hours = $810.00	$810.00
AWS Lambda	10,000 requests per month with 512MB memory, 2-second average duration (Requests: 10,000 requests, Compute: 10,000 requests × 2s × 0.5GB = 10,000 GB-seconds)	$0.0000002 × 10,000 requests + $0.0000166667 × 10,000 GB-seconds = $0.002 + $0.167 = $0.169	$0.17
	5GB storage for documents,		

Amazon S3	1,000 PUT requests, 10,000 GET requests per month (Storage: 5 GB, Put Requests: 1,000 requests, Get Requests: 10,000 requests)	$0.023 × 5GB + $0.005 × 1 + $0.0004 × 10 = $0.115 + $0.005 + $0.004 = $0.124	$0.12
Amazon DynamoDB	On-demand mode: 1,000 read requests, 500 write requests per month, 1GB storage (Read Requests: 1,000 requests, Write Requests: 500 requests, Storage: 1 GB)	$0.125/1M × 0.001M reads + $1.25/1M × 0.0005M writes + $0.25 × 1GB = $0.000125 + $0.000625 + $0.25 = $0.25	$0.25
Amazon API Gateway	10,000 HTTP API requests per month (Requests: 10,000 requests)	$1.00/1M × 0.01M requests = $0.01	$0.01
Total	**All services**	**Sum of all calculations**	**$810.67/month**

Free Tier

Free tier information by service: - **Amazon Bedrock (Claude 3.5 Haiku)**: No free tier for Bedrock foundation models - **Amazon Kendra (Developer Edition)**: No free tier for Kendra - **AWS Lambda**: First 12 months: 1M requests/month and 400,000 GB-seconds/month free - **Amazon S3**: First 12 months: 5GB storage, 20,000 GET requests, 2,000 PUT requests free - **Amazon DynamoDB**: Always free: 25GB storage, 25 read capacity units, 25 write capacity units - **Amazon API Gateway**: First 12 months: 1M API calls per 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 Bedrock (Claude 3.5 Haiku)	$0/month	$0/month	$0/month
Amazon Kendra (Developer Edition)	$405/month	$810/month	$1620/month
AWS Lambda	$0/month	$0/month	$0/month
Amazon S3	$0/month	$0/month	$0/month
Amazon DynamoDB	$0/month	$0/month	$0/month
Amazon API Gateway	$0/month	$0/month	$0/month

Key Cost Factors

- **Amazon Bedrock (Claude 3.5 Haiku)**: Processing 100K input tokens and 50K output tokens per month
- **Amazon Kendra (Developer Edition)**: 24/7 operation for development and testing
- **AWS Lambda**: 10,000 requests per month with 512MB memory, 2-second average duration
- **Amazon S3**: 5GB storage for documents, 1,000 PUT requests, 10,000 GET requests per month
- **Amazon DynamoDB**: On-demand mode: 1,000 read requests, 500 write requests per month, 1GB storage
- **Amazon API Gateway**: 10,000 HTTP API requests per month

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 Bedrock (Claude 3.5 Haiku)	$0.12
Amazon Kendra (Developer Edition)	$810.00
AWS Lambda	$0.17
Amazon S3	$0.12
Amazon DynamoDB	$0.25
Amazon API Gateway	$0.01
Total Monthly Cost	**$810**

Growth Pattern	Month 1	Month 3	Month 6	Month 12
Steady	$810/mo	$810/mo	$810/mo	$810/mo

Moderate	$810/mo	$893/mo	$1034/mo	$1386/mo
Rapid	$810/mo	$980/mo	$1305/mo	$2312/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 monitoring and logging costs
- Development and maintenance costs
- Custom domain and SSL certificate costs
- Backup and disaster recovery costs
- Third-party integrations or tools
- Network infrastructure costs (VPC, NAT Gateway, etc.)
- Security services (WAF, Shield, etc.)

Recommendations

Immediate Actions

- Consider using Kendra Developer Edition only during active development to reduce costs
- Implement efficient caching strategies to reduce Bedrock token usage
- Optimize Lambda function memory allocation based on actual performance requirements
- Use S3 Intelligent Tiering for document storage if access patterns vary
- Monitor DynamoDB usage and consider provisioned capacity if usage becomes predictable #### Best Practices
- Set up billing alerts and cost monitoring dashboards
- Implement proper error handling to avoid unnecessary retries and costs
- Use AWS Cost Explorer to track spending trends
- Consider implementing usage quotas to prevent unexpected cost spikes
- Regularly review and optimize resource configurations

Cost Optimization Recommendations

Immediate Actions

- Consider using Kendra Developer Edition only during active development to reduce costs
- Implement efficient caching strategies to reduce Bedrock token usage
- Optimize Lambda function memory allocation based on actual performance requirements

Best Practices

- Set up billing alerts and cost monitoring dashboards
- Implement proper error handling to avoid unnecessary retries and costs
- Use AWS Cost Explorer to track spending trends

Conclusion

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