Intelligent Document Processing Application Cost Analysis Estimate Report

Service Overview

Intelligent Document Processing Application 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 Sonnet model for document processing ($3.00 per million input tokens, $15.00 per million output tokens)
- Average document size of 2MB with 2,000 input tokens and 500 output tokens per document
- Lambda functions with 512MB memory allocation and 3-second average execution time
- DynamoDB on-demand billing with standard read/write patterns
- S3 Standard storage class for document storage
- API Gateway REST API for document upload and retrieval endpoints
- No caching or optimization applied initially
- Documents retained for 30 days in S3 for audit purposes

Limitations and Exclusions

- Data transfer costs between regions (single region deployment)
- Custom model training or fine-tuning costs
- Development and maintenance costs
- Third-party integrations or external API costs
- CloudWatch detailed monitoring costs beyond basic logs
- VPC or networking costs (using default VPC)
- Backup and disaster recovery costs
- Compliance and security audit costs

Cost Breakdown

Unit Pricing Details

No detailed unit pricing information available.

Cost Calculation

Service	Usage	Calculation	Monthly Cost
Amazon Bedrock (Claude 3.5 Sonnet)		N/A	N/A
AWS Lambda		N/A	N/A
Amazon S3		N/A	N/A
Amazon DynamoDB		N/A	N/A
Amazon API Gateway		N/A	N/A
AWS CloudWatch		N/A	N/A

Free Tier

Free tier information by service: - **Amazon Bedrock (Claude 3.5 Sonnet)**: No free tier available for Bedrock foundation models - **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/write capacity units - **Amazon API Gateway**: First 12 months: 1M API calls/month free - **AWS CloudWatch**: 5GB log ingestion, 10 custom metrics free per month

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 Sonnet)	Varies	Varies	Varies
AWS Lambda	Varies	Varies	Varies
Amazon S3	Varies	Varies	Varies
Amazon DynamoDB	Varies	Varies	Varies
Amazon API Gateway	Varies	Varies	Varies
AWS CloudWatch	Varies	Varies	Varies

Key Cost Factors

- Request volume and frequency
- Data storage requirements
- Data transfer between services
- Compute resources utilized

Projected Costs Over Time

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

Insufficient data to generate cost projections. See Custom Analysis Data section for available cost information.

Detailed Cost Analysis

Pricing Model

ON DEMAND

Exclusions

- Data transfer costs between regions (single region deployment)
- Custom model training or fine-tuning costs
- Development and maintenance costs
- Third-party integrations or external API costs
- CloudWatch detailed monitoring costs beyond basic logs
- VPC or networking costs (using default VPC)
- Backup and disaster recovery costs
- Compliance and security audit costs

Total Monthly Costs

Key	Value
Low Usage	See nested table below
Medium Usage	See nested table below
High Usage	See nested table below

Low Usage

Key	Value
Bedrock	$13.50
Lambda	$0.15
S3	$0.05
Dynamodb	$0.00
Api Gateway	$0.02
Cloudwatch	$2.00
Total	$15.72

Medium Usage

Key	Value
Bedrock	$135.00
Lambda	$1.51
S3	$0.47
Dynamodb	$0.02

Api Gateway	$0.21
Cloudwatch	$8.00
Total	$145.21

High Usage

Key	Value
Bedrock	$675.00
Lambda	$7.56
S3	$2.32
Dynamodb	$0.11
Api Gateway	$1.05
Cloudwatch	$25.00
Total	$711.04

Recommendations

Immediate Actions

- Optimize prompt engineering to reduce token usage for Claude 3.5 Sonnet
- Implement response caching for common document types to reduce Bedrock API calls
- Use S3 Intelligent Tiering for documents older than 30 days
- Configure Lambda memory allocation based on actual usage patterns
- Enable DynamoDB auto-scaling for variable workloads
- Implement API Gateway caching for frequently accessed endpoints #### Best Practices
- Monitor token usage patterns and adjust model selection accordingly
- Implement proper error handling to avoid unnecessary retries and costs
- Use AWS Cost Explorer to track spending trends across services
- Set up billing alerts for proactive cost management
- Regular review of CloudWatch logs retention and metrics usage
- Consider using AWS Budgets for cost control and forecasting

Cost Optimization Recommendations

Immediate Actions

- Optimize prompt engineering to reduce token usage for Claude 3.5 Sonnet

- Implement response caching for common document types to reduce Bedrock API calls
- Use S3 Intelligent Tiering for documents older than 30 days

Best Practices

- Monitor token usage patterns and adjust model selection accordingly
- Implement proper error handling to avoid unnecessary retries and costs
- Use AWS Cost Explorer to track spending trends across services

Conclusion

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