AWS Pricing Analysis: Intelligent Document Processing Application

Executive Summary

This document provides a comprehensive cost analysis for the Intelligent Document Processing (IDP) application built on AWS serverless architecture. The application processes documents through a three-stage pipeline using Amazon Bedrock's Claude 4 model for OCR, classification, and summarization tasks.

Key Cost Drivers: - Amazon Bedrock Foundation Models (Claude 4): Primary cost component - AWS Lambda: Compute for orchestration and API handling - Amazon API Gateway: RESTful API endpoints - Amazon DynamoDB: Results storage - Amazon S3: Document storage - AWS Step Functions: Pipeline orchestration

Architecture Overview

The application follows a serverless event-driven architecture:

1. **Frontend**: React application for document upload and results display
2. **API Layer**: API Gateway with Lambda functions for upload/results handling
3. **Processing Pipeline**: Step Functions orchestrating three sequential tasks
4. **AI Processing**: Amazon Bedrock Claude 4 for OCR, classification, and summarization
5. **Storage**: S3 for documents, DynamoDB for results and metadata

Service-by-Service Pricing Analysis

1. Amazon Bedrock Foundation Models (Claude 4)

Primary Cost Component - Estimated 70-80% of total costs

Claude 3.5 Haiku (Recommended for cost optimization)

- **Input Tokens**: $0.80 per million tokens
- **Output Tokens**: $4.00 per million tokens

Claude 3.5 Sonnet (Balanced performance/cost)

- **Input Tokens**: $3.00 per million tokens
- **Output Tokens**: $15.00 per million tokens

Claude Sonnet 4 (Premium option)

- **Input Tokens**: $3.00 per million tokens

- **Output Tokens**: $15.00 per million tokens

Token Estimation per Document: - **OCR Task**: 2,000-8,000 input tokens, 500-2,000 output tokens - **Classification Task**: 1,000-3,000 input tokens, 50-200 output tokens
- **Summarization Task**: 1,500-4,000 input tokens, 200-800 output tokens

2. AWS Lambda

Request Pricing: $0.0000002 per request **Compute Pricing**: - Tier 1 (0-6B GB-seconds): $0.0000166667 per GB-second - Tier 2 (6B-15B GB-seconds): $0.0000150000 per GB-second - Tier 3 (15B+ GB-seconds): $0.0000133334 per GB-second

Estimated Lambda Functions: - Upload Handler: 512MB, 2-3 seconds per execution - Results Handler: 256MB, 1-2 seconds per execution - Pipeline Orchestrator: 512MB, 1-2 seconds per execution - OCR Processor: 1024MB, 5-15 seconds per execution - Classification Processor: 512MB, 2-5 seconds per execution - Summarization Processor: 512MB, 3-8 seconds per execution

3. Amazon API Gateway

REST API Pricing (Tiered): - First 333 million requests/month: $3.50 per million requests - Next 667 million requests/month: $2.80 per million requests - Next 19 billion requests/month: $2.38 per million requests - Over 20 billion requests/month: $1.51 per million requests

HTTP API Pricing (Lower cost alternative): - First 300 million requests/month: $1.00 per million requests - Over 300 million requests/month: $0.90 per million requests

4. Amazon DynamoDB

On-Demand Pricing: - **Read Capacity**: $0.00013 per hour per unit (beyond 25 free units) - **Write Capacity**: $0.00065 per hour per unit (beyond 25 free units) - **Storage**: $0.25 per GB-month (beyond 25 GB free tier)

Free Tier: 25 GB storage, 25 read/write capacity units per month

5. Amazon S3

Standard Storage: - First 50 TB/month: $0.023 per GB - Next 450 TB/month: $0.022 per GB - Over 500 TB/month: $0.021 per GB

Request Pricing: - PUT/COPY/POST/LIST: $0.005 per 1,000 requests - GET and other requests: $0.0004 per 1,000 requests

6. AWS Step Functions

Standard Workflows: $0.025 per 1,000 state transitions **Express Workflows**: $1.00 per million requests + $0.00001667 per GB-second

Usage Scenarios and Cost Projections

Scenario 1: Low Usage (100 documents/month)

Monthly Processing Volume: 100 documents **Average Document**: 5 pages, mixed PDF/images

Bedrock Costs (Claude 3.5 Haiku):

- Input tokens: 100 docs × 4,500 avg tokens = 450K tokens
- Output tokens: 100 docs × 1,000 avg tokens = 100K tokens
- **Cost**: (450K × $0.80/M) + (100K × $4.00/M) = $0.36 + $0.40 = **$0.76**

Lambda Costs:

- Total executions: 600 (6 functions × 100 docs)
- Compute time: ~1,800 GB-seconds
- **Cost**: (600 × $0.0000002) + (1,800 × $0.0000166667) = **$0.03**

API Gateway Costs:

- API calls: ~400 requests (upload + polling)
- **Cost**: 400 × $3.50/M = **$0.001**

DynamoDB Costs:

- Storage: ~0.1 GB (within free tier)
- Read/Write operations: minimal (within free tier)
- **Cost**: **$0.00**

S3 Costs:

- Storage: ~1 GB documents
- Requests: ~200 PUT/GET requests
- **Cost**: (1 × $0.023) + (200 × $0.000005) = **$0.024**

Step Functions Costs:

- State transitions: 300 (3 per document)
- **Cost**: 300 × $0.025/1000 = **$0.008**

Total Monthly Cost: ~$0.82

Scenario 2: Medium Usage (1,000 documents/month)

Monthly Processing Volume: 1,000 documents

Bedrock Costs (Claude 3.5 Haiku):

- Input tokens: 4.5M tokens
- Output tokens: 1M tokens
- **Cost**: (4.5M × $0.80/M) + (1M × $4.00/M) = $3.60 + $4.00 = **$7.60**

Lambda Costs:

- Total executions: 6,000
- Compute time: ~18,000 GB-seconds
- **Cost**: (6,000 × $0.0000002) + (18,000 × $0.0000166667) = **$0.30**

API Gateway Costs:

- API calls: ~4,000 requests
- **Cost**: 4,000 × $3.50/M = **$0.014**

DynamoDB Costs:

- Storage: ~1 GB (within free tier)
- Operations: moderate (likely within free tier)
- **Cost**: **$0.00**

S3 Costs:

- Storage: ~10 GB documents
- Requests: ~2,000 requests
- **Cost**: (10 × $0.023) + (2,000 × $0.000005) = **$0.24**

Step Functions Costs:

- State transitions: 3,000
- **Cost**: 3,000 × $0.025/1000 = **$0.075**

Total Monthly Cost: ~$8.23

Scenario 3: High Usage (10,000 documents/month)

Monthly Processing Volume: 10,000 documents

Bedrock Costs (Claude 3.5 Haiku):

- Input tokens: 45M tokens
- Output tokens: 10M tokens
- **Cost**: (45M × $0.80/M) + (10M × $4.00/M) = $36.00 + $40.00 = **$76.00**

Lambda Costs:

- Total executions: 60,000
- Compute time: ~180,000 GB-seconds
- **Cost**: (60,000 × $0.0000002) + (180,000 × $0.0000166667) = **$3.01**

API Gateway Costs:

- API calls: ~40,000 requests
- **Cost**: 40,000 × $3.50/M = **$0.14**

DynamoDB Costs:

- Storage: ~10 GB
- Operations: higher volume, some paid usage
- **Cost**: **$2.50** (estimated)

S3 Costs:

- Storage: ~100 GB documents
- Requests: ~20,000 requests
- **Cost**: (100 × $0.023) + (20,000 × $0.000005) = **$2.40**

Step Functions Costs:

- State transitions: 30,000
- **Cost**: 30,000 × $0.025/1000 = **$0.75**

Total Monthly Cost: ~$84.80

Cost Optimization Recommendations

1. Model Selection Strategy

- **Start with Claude 3.5 Haiku** for cost optimization (5x cheaper than Sonnet)
- **Upgrade to Claude 3.5 Sonnet** only if accuracy requirements demand it
- **Monitor token usage** and optimize prompts to reduce input/output tokens

2. Architecture Optimizations

- **Use HTTP API Gateway** instead of REST API (50% cost reduction)
- **Implement response caching** in API Gateway to reduce Lambda invocations
- **Optimize Lambda memory allocation** based on actual usage patterns
- **Use DynamoDB on-demand billing** for variable workloads

3. Processing Optimizations

- **Batch processing** for multiple documents to reduce per-document overhead
- **Implement document preprocessing** to reduce token consumption
- **Use Step Functions Express Workflows** for high-volume scenarios
- **Optimize prompt engineering** to minimize token usage

4. Storage Optimizations

- **Use S3 Intelligent Tiering** for long-term document storage
- **Implement lifecycle policies** to move old documents to cheaper storage classes
- **Compress documents** before storage when possible

5. Monitoring and Alerting

- **Set up CloudWatch billing alerts** for cost thresholds
- **Monitor token usage patterns** to identify optimization

opportunities
- **Track processing times** to optimize Lambda configurations
- **Implement cost allocation tags** for detailed cost tracking

Free Tier Benefits

First 12 Months (New AWS Accounts): - **Lambda**: 1M requests + 400,000 GB-seconds per month - **API Gateway**: 1M API calls per month - **DynamoDB**: 25 GB storage + 25 read/write capacity units - **S3**: 5 GB standard storage + 20,000 GET + 2,000 PUT requests

Always Free: - **DynamoDB**: 25 GB storage + 25 read/write capacity units per month - **Lambda**: 1M requests + 400,000 GB-seconds per month

Scaling Considerations

Linear Scaling Components

- **Bedrock costs** scale directly with document volume and complexity
- **Lambda execution costs** scale with processing volume
- **Step Functions costs** scale with workflow executions

Economies of Scale

- **API Gateway** pricing decreases with higher volume tiers
- **S3 storage** costs decrease with higher usage tiers
- **Lambda compute** costs decrease with higher usage tiers

Break-even Analysis

- **Low volume** (< 500 docs/month): Primarily Bedrock costs
- **Medium volume** (500-5,000 docs/month): Balanced across services
- **High volume** (> 5,000 docs/month): Infrastructure costs become significant

Risk Factors and Mitigation

Cost Risks

1. **Unexpected token usage spikes** from complex documents
2. **API Gateway costs** from excessive polling
3. **DynamoDB hot partitions** causing throttling and retries

Mitigation Strategies

1. **Implement token usage monitoring** and alerts
2. **Use WebSocket connections** for real-time updates instead of polling
3. **Design DynamoDB schema** with proper partition key

distribution
4. **Set up AWS Budgets** with automatic actions

Conclusion

The Intelligent Document Processing application's costs are primarily driven by Amazon Bedrock usage, which typically represents 70-80% of total expenses. For typical usage scenarios:

- **Low usage (100 docs/month)**: ~$0.82/month
- **Medium usage (1,000 docs/month)**: ~$8.23/month
- **High usage (10,000 docs/month)**: ~$84.80/month

The serverless architecture provides excellent cost efficiency for variable workloads, with most infrastructure costs remaining minimal compared to AI processing costs. Key optimization opportunities lie in model selection, prompt engineering, and token usage optimization.

Recommended Starting Configuration: - Claude 3.5 Haiku for cost optimization - HTTP API Gateway for reduced API costs - DynamoDB on-demand billing - S3 Standard storage with lifecycle policies - CloudWatch monitoring and billing alerts

This configuration provides a cost-effective foundation that can scale efficiently with business growth while maintaining the flexibility to optimize based on actual usage patterns.

Analysis Date: December 1, 2025
Region: US East (N. Virginia)
Currency: USD
Pricing: Current AWS On-Demand rates