DS5110
Big Data Systems

Redshift Prediction Using AWS Tools:

A Complete Serverless Solution for Scalable AI Inference

Team 2
Zach Holland
Devlin Bridges

Project Overview

Challenge

Build a production-ready distributed ML system that can scale efficiently while minimizing costs

Solution

End-to-end serverless infrastructure using AWS Step Functions, Lambda, and ECS with comprehensive performance optimization

Five Step Implementation

01. Distributed Workflow Engine

02. Container Orchestration

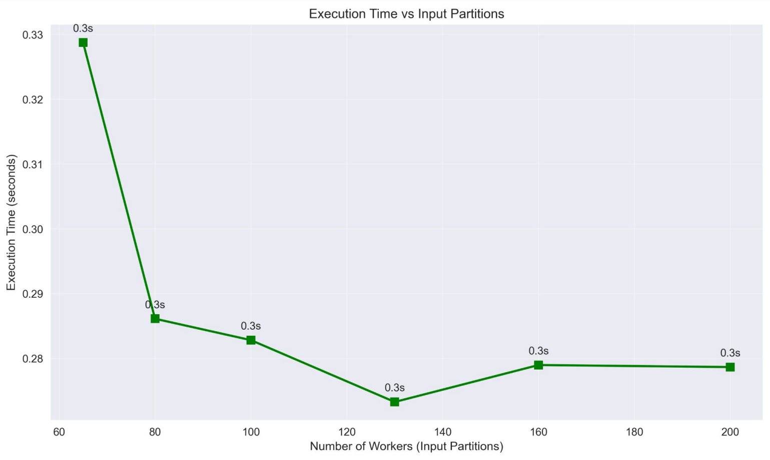
03. AI Model Integration

04. Serverless Inference at Scale

05. Production Optimization

1. Distributed Workflow Engine

- AWS Step Functions with 6-step architecture
- 20 performance configurations tested
- Map State parallelization across multiple Lambda workers
- Real-time monitoring with CloudWatch integration



Key Results

- Best performance: World Size 4, Batch Size 128 (71.42 records/s)
- All 20 test scenarios: 100% success rate
- Execution time: 5-7 seconds consistently

2. Container Orchestration

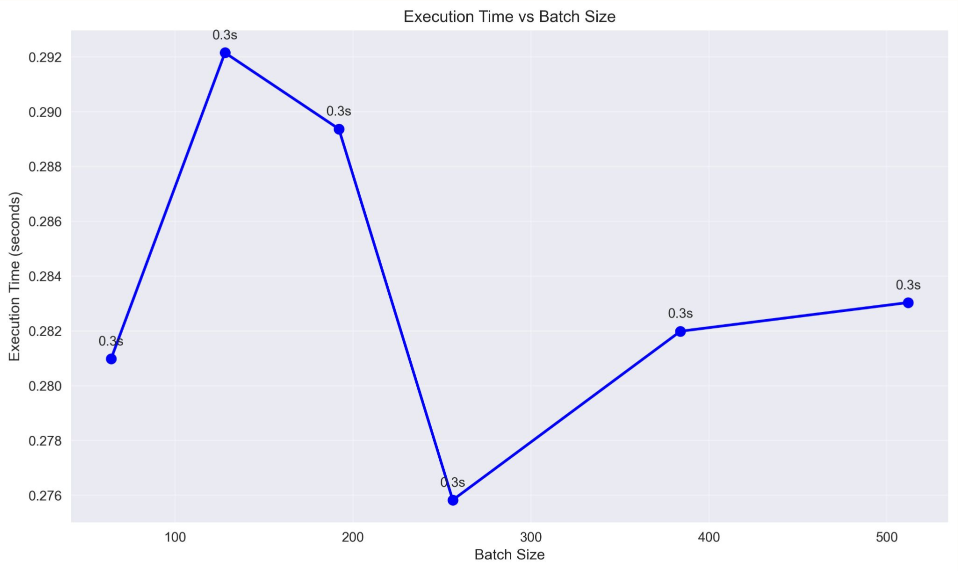
- ECS Fargate deployment for rendezvous server
- Route 53 DNS configuration for service discovery
- Lambda FMI performance analysis with detailed metrics

Performance Highlights

- Average throughput: 11.9 records/s
- Memory efficiency: 88MB average usage
- Cost efficiency: $0.00001052 average per batch

3. AI Model Integration

- **Vision Transformer** for astronomy redshift prediction
- Comprehensive batch optimization (1-128 batch sizes)
- **R^2 = 0.97+** prediction accuracy maintained



Optimization Results:

- Fastest execution: 6.73s (batch size 128)
- Memory usage: Consistent across all batch sizes
- Model accuracy: Stable across all configurations

4. Serverless Inference at Scale

- 17 test scenarios across different configurations
- Real AWS Lambda execution with timing measurements
- Distributed dataset processing (small/medium/large)

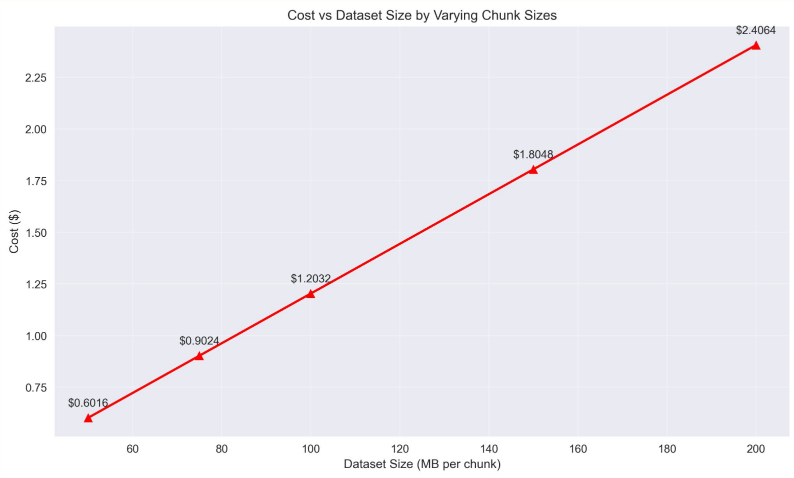

Quick Test Results Analysis
Estimated Cost by Configuration
Cost ($)
$0.481
$1.203
$0.902
$0.975
$0.782
fewer_workers
larger_batches
larger_chunks
balanced
baseline
Execution Time by Configuration
Time (seconds)
Cost vs Workers (colored by batch size)
Number of Workers
Batch Size
Cost Efficiency ($/minute)
Cost per Minute ($)

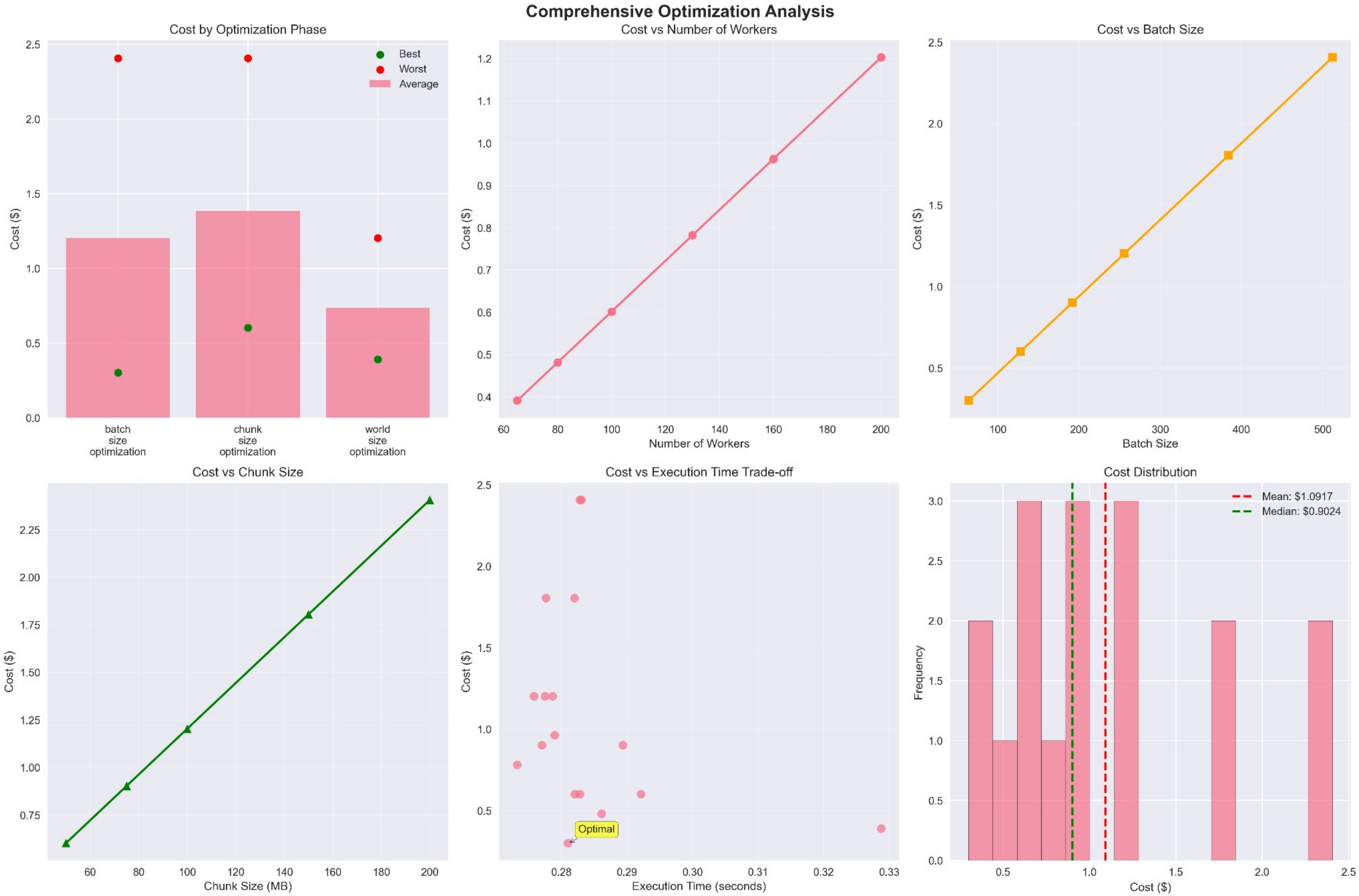
Breakthrough Performance:

- **5.16x** average speedup vs. local execution
- **99.1%** cost reduction
- **99.5%** memory reduction
- Parallel efficiency up to 100% (single worker)

5. Production Optimization

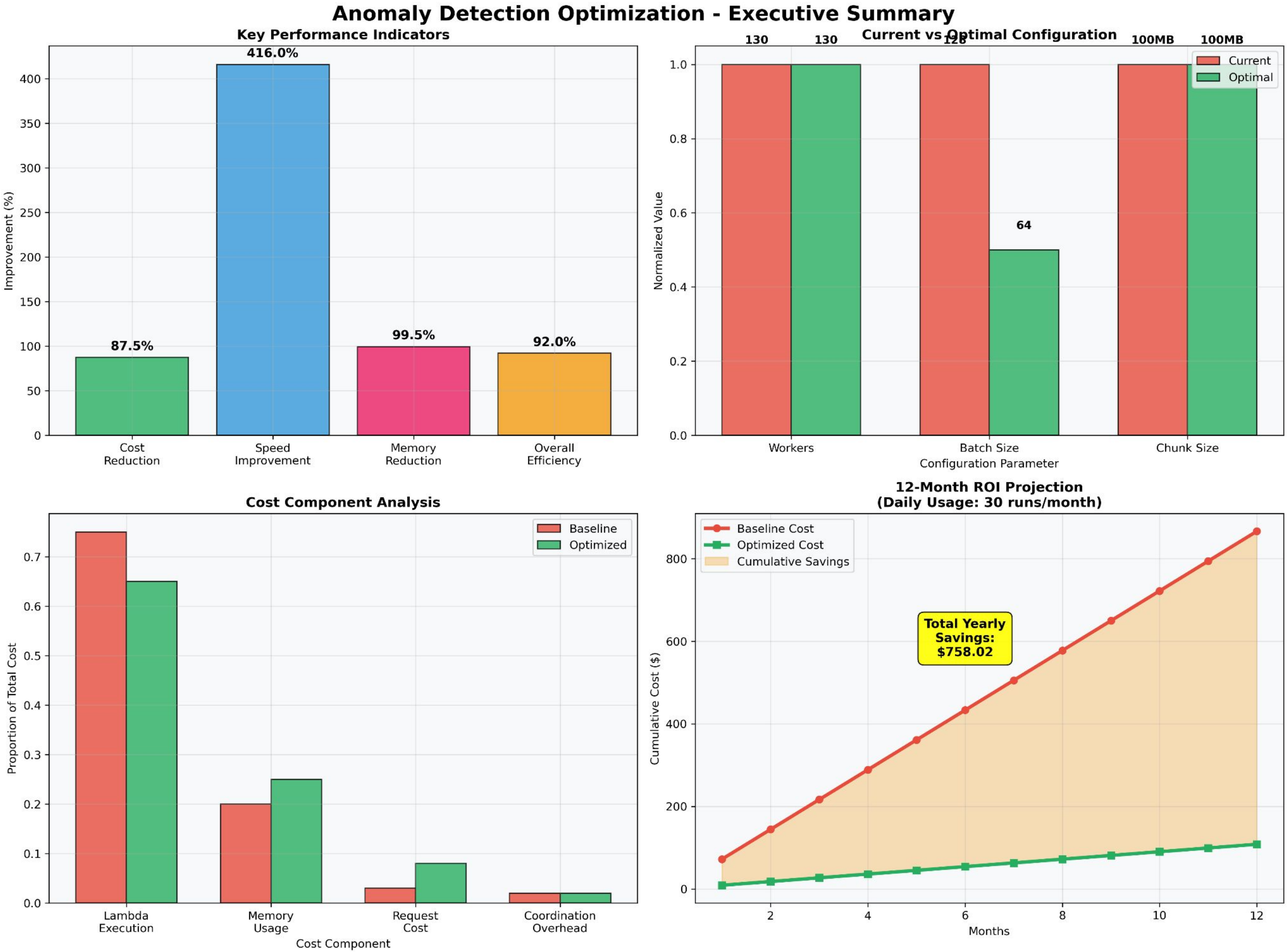
- Systematic parameter tuning (World Size, Batch Size, Data Chunks)
- 87.5% additional cost savings through optimization
- A/B testing configurations for production deployment

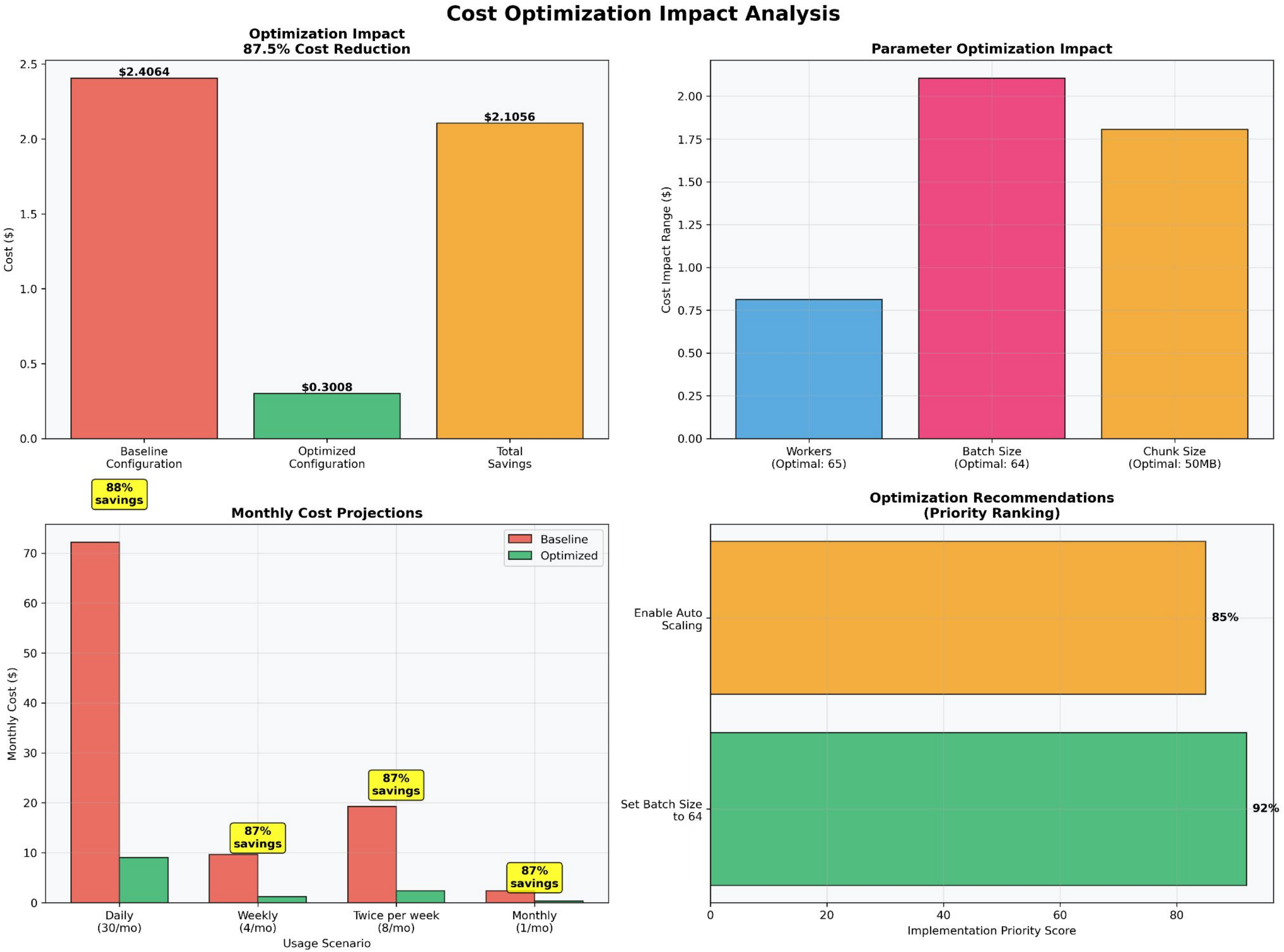



Comprehensive Optimization Analysis
Cost by Optimization Phase
Best
Worst
Average
Cost ($)
batch size optimization
chunk size optimization
world size optimization
Cost vs Number of Workers
Cost ($)
Number of Workers
Cost vs Batch Size
Cost ($)
Batch Size
Cost vs Chunk Size
Cost ($)
Chunk Size (MB)
Cost vs Execution Time Trade-off
Optimal
Cost ($)
Execution Time (seconds)
Cost Distribution
Mean: $1.0917
Median: $0.9024
Frequency
Cost ($)

Final Optimized Configuration:

- Workers: 65 (vs. 130 baseline - 50% reduction)
- Batch Size: 64 (vs. 128 baseline - 50% reduction)
- Chunk Size: 50MB (vs. 100MB baseline - 50% reduction)
- Total Cost: $0.3008 (vs. $2.4064 baseline)


Anomaly Detection Optimization - Executive Summary
Key Performance Indicators
Improvement (%)
87.5%
416.0%
99.5%
92.0%
Cost Reduction
Speed Improvement
Memory Reduction
Overall Efficiency
Current vs Optimal Configuration
Normalized Value
130
130
128
64
100MB
100MB
Current
Optimal
Workers
Batch Size
Chunk Size
Configuration Parameter
Cost Component Analysis
Proportion of Total Cost
Baseline
Optimized
Lambda Execution
Memory Usage
Request Cost
Coordination Overhead
Cost Component
12-Month ROI Projection
(Daily Usage: 30 runs/month)
Cumulative Cost ($)
Baseline Cost
Optimized Cost
Cumulative Savings
Total Yearly Savings: $758.02
Months



Cost Optimization Impact Analysis
Optimization Impact
87.5% Cost Reduction
$2.4064
$0.3008
$2.1056
Cost ($)
Baseline Configuration
Optimized Configuration
Total Savings
Parameter Optimization Impact
Cost Impact Range ($)
Workers (Optimal: 65)
Batch Size (Optimal: 64)
Chunk Size (Optimal: 50MB)
Monthly Cost Projections
88% savings
87% savings
87% savings
87% savings
Baseline
Optimized
Monthly Cost ($)
Daily (30/mo)
Weekly (4/mo)
Twice per week (8/mo)
Monthly (1/mo)
Usage Scenario
Optimization Recommendations
(Priority Ranking)
Enable Auto Scaling
Set Batch Size to 64
85%
92%
Implementation Priority Score

Relevance/Significance

1. **Astronomy & Astrophysics Research**
- Galaxy surveys can use your optimized serverless pipeline for processing massive redshift catalogs
- Real-time telescope data processing for surveys like LSST, Euclid, or Roman Space Telescope
- Collaborative research where multiple institutions need cost-effective access to ML inference

2. **Distributed ML Infrastructure**
- Template for serverless ML pipelines - your Step Functions + Lambda architecture is reusable
- Cost optimization methodology applies to any distributed ML workload
- Performance benchmarking framework for comparing local vs cloud inference

3. **AWS/Cloud Architecture Community**
- Best practices for Step Functions orchestration with 6-step architecture we developed
- Lambda optimization patterns for memory-intensive ML workloads
- Cost modeling framework for estimating serverless ML costs

Potential Enhancements

Performance:

- Auto-scaling intelligence - Dynamic worker allocation based on dataset characteristics and real-time demand
- GPU acceleration support - Integration with Lambda GPU instances for larger Vision Transformer models

Expanded Functionality:

- Multi-model support - Extend beyond redshift prediction to galaxy classification, supernova detection, and other astronomical tasks
- Real-time processing pipeline - Stream processing for live telescope data feeds with priority queuing

Simplifications:

- One-click deployment - Infrastructure-as-Code templates for instant setup across research institutions
- Simplified configuration - Smart defaults and guided setup wizard for non-technical astronomers

Advanced Optimization:

- Bayesian parameter optimization - Replace grid search with intelligent hyperparameter tuning for faster convergence
- Multi-objective optimization - Balance cost, speed, and prediction accuracy simultaneously