

Predictive ML-Based Autoscaling

for Containerized Microservices (ECS Fargate)

Multi-Signal Ensemble with Formal Guarantees

Marcin Wypyszynski marcin.wypyszynski@gmail.com

Cloud & Navigation Platform Engineering | March 2026 | Full paper at wypyszynski.dev

Problem

AWS ECS threshold-based autoscaling operates reactively, introducing a 3–7 minute lag between load increase and capacity. This causes service degradation during traffic spikes for containerized microservices.

Solution: 6-Component ML Pipeline

A predictive system making scaling decisions **2–5 minutes before** demand spikes:

1. **Agreement-Based Ensemble Predictor** – Adaptive exponential smoothing (CV-driven $\alpha \in [0.1, 0.6]$) combined with weighted moving average; divergence-based estimator selection
2. **Linear Trend Extrapolator** – EMA-smoothed velocity eliminates parabolic overshoot of kinematic models
3. **Multi-Signal Leading Indicators** – Online-learned request-to-CPU elasticity ($E = \text{Cov}/\text{Var}$) for ALB; correlation-learned weights for Micrometer JVM metrics
4. **Percentile-Based Burst Detector** – P95 + median deviation with capped $1.5\times$ scale factor
5. **Pattern Learner** – Hourly/daily temporal load profiles with preemptive scaling triggers
6. **Confidence-Aware Hysteresis Controller** – Prediction confidence modulates thresholds; provable oscillation bounds

Key Innovations

- **Minimum baseline stability guards** replacing naïve $\epsilon = 10^{-10}$ denominators – eliminates false signals at low utilization
- **Online correlation learning** for both ALB elasticity and Micrometer signal weights
- **Formal safety guarantees** – proven bounded scaling invariant ($N_{\min} \leq N^* \leq N_{\max}$) for all execution paths
- **Oscillation bound theorem** – $\max \lfloor T/(K_{\text{consec}} + 1) \rfloor$ oscillations in T evaluation periods
- **9-level decision hierarchy** – burst $\rightarrow$ queue overload $\rightarrow$ leading indicators $\rightarrow$ prediction $\rightarrow$ trend $\rightarrow$ pattern $\rightarrow$ reactive $\rightarrow$ scale-down $\rightarrow$ no change

Production Results (6 ECS Services, 2 Environments)

Metric	Result
Lambda invocations (verification window)	96
Lambda errors	0 (0%)
Average decision latency	< 400 ms (full ML pipeline)
ML analysis time	5–10 ms
CloudWatch metrics published/invoke	19
Unintended ECS scaling events	0
Safety clamp enforcements	Validated (enforce_min)

Technology Stack

AWS Lambda (Python 3.11) ECS Fargate CloudWatch DynamoDB EventBridge ALB Micrometer

Managed Services

Service	Tasks	Target CPU	Type
Geospatial Search	2–8	65%	Text search API
Charger Station Lookup	1–5	70%	EV charger finder
Dealership Finder	1–4	70%	Dealer locator

© 2026 Marcin Wypyszynski. Full paper with mathematical proofs, theorems, algorithms, and complexity analysis available at wypyszynski.dev.