

Predictive Infrastructure Scaling at Microsoft: Stamp Right-Sizing

Author: Aeryn Light

Role: Software Development Engineer, Microsoft

Timeframe: 2021–2024

Overview

At Microsoft, I led the design and implementation of a proactive infrastructure service known as the **Stamp Right-Sizing Service**, which dramatically reduced outages across our cloud platform. By analyzing resource usage trends across virtual machine “stamps” (regional clusters of virtual machines), we were able to scale capacity before customer impact occurred.

The Problem

Each month, our broader organization experienced up to **10 outages**, most of which stemmed from inadequate capacity on cloud stamps. These outages required urgent, reactive scaling, often performed manually and under pressure by on-call engineers.

Reactive scaling had two major drawbacks:

- **Customer impact** due to delayed incident response.
- **Operational burden** on on-call staff who had to manually interpret telemetry and apply fixes.

My Solution

I created a service that:

- Ran **daily** to check capacity across all regional stamps.
- Used telemetry to identify under-provisioned resources.
- Generated **proactive incidents** with detailed, tailored instructions for scaling.
- Included **edge case handling**, fallback logic, and guidance for atypical situations.

To reduce alert fatigue and noise, I introduced a **moving average model** that prioritized recent usage spikes without overreacting to short-term volatility. I also built a **standalone testing tool** that mirrored the service logic, enabling safe iteration and validation using historical incident data.

Results

- **Outages dropped** from 10/month to **2/month**.
- The **on-call load decreased** from 80+ manual capacity changes per month to just 20.

This service is now considered a foundational component of our infrastructure health suite.

Technical Highlights

- **Tech stack:** C#, .NET, Azure, MongoDB, AKS, KQL
- **Features:** Forecasting logic, incident templating, standalone CLI for testing, scalable telemetry queries
- **Design principle:** Prioritize **maintainability, signal over noise,** and **user confidence** during incidents

Reflections

Through this project, I gained a deeper understanding of:

- How to balance long-term trend modeling with short-term risk mitigation.
- The importance of **self-documenting systems** in distributed environments.
- How thoughtful tooling design can transform the on-call experience from reactive firefighting into sustainable operations.