SRE Implementation

for a Global E-commerce Platform

Overview

This case study highlights the implementation of Site Reliability Engineering (SRE) for a global e-commerce platform. By adopting a hybrid SRE model, the team focused on proactive monitoring, data-driven scaling, and automation to handle traffic spikes, optimize costs, and ensure high system reliability. The model allowed the platform to scale efficiently, especially during peak sales events. This approach resulted in significant annual savings, improved performance, and minimized downtime, all while fostering a culture of shared responsibility for system reliability.

Client Profile

Our client is one of Asia's largest clothing retailers with more than 2,500 stores across the globe. The company operates in segments such as manufacturing and sale of apparel in the domestic and overseas markets.

Business Challenges

Managing Traffic Spikes Without Overspending

- Sales Events and Traffic Surges: Large-scale sales events like Black Friday and Cyber Monday caused huge traffic spikes.
- Costly Infrastructure Scaling: Keeping the platform continuously scaled up to handle traffic surges 24/7 was cost-prohibitive.

Ensuring Fast Performance

- Customer Experience: Even a slight delay of half a second resulted in customers abandoning their carts.
- Slow Websites: Slow load times not only impacted sales but also customer trust.

Avoiding Downtime

- Impact of Downtime: Even brief downtime led to significant revenue loss and poor customer satisfaction.

- Uptime Goal: Achieving a 99.999% uptime while balancing cost efficiency was a significant challenge.

Team Collaboration and Ownership

- Developer Focus: Developers were primarily focused on feature delivery, leaving system reliability to be reactive rather than proactive.
- IT Team Limitations: Traditional IT teams focused on fixing issues post-failure, creating silos and gaps in system reliability.

Business Requirement

- Scalable, cost-effective infrastructure to manage traffic spikes and seasonal events
- High performance and 99% uptime to prevent revenue loss and customer dissatisfaction
- A proactive SRE strategy with continuous monitoring, optimization, and collaborative ownership of system reliability between developers and operations

Choosing the Right SRE Model

After evaluating several Site Reliability Engineering models, we selected a hybrid approach:

- **Dedicated SRE Team:** Could focus on reliability but might face scaling issues.
- **Consulting SRE:** Would guide but needed strong collaboration with developers.
- **Hybrid Team (Chosen Model):** A central SRE team alongside developer representatives in each microservice team. This hybrid approach integrated developers and SREs, sharing ownership of both feature development and system reliability.

Understanding SLIs, SLOs, and SLAs for Reliable Service Performance

To ensure a seamless user experience, SLIs, SLOs, and SLAs work together to maintain service quality:

1. SLIs (Service Level Indicators) that measure system performance include:

- **Availability:** 99.95% uptime
- **Latency:** Pages load within 2-3 seconds for 95% of requests
- **Error Rate:** Less than 0.1% failure rate
- **Throughput:** Orders processed per second
- **Checkout Success:** Success rate above 99.5%

2. SLOs (Service Level Objectives) define performance goals to exceed SLAs:

- **Availability:** 99.98% uptime with a buffer above SLA commitment
- **Latency:** 95% of product pages load within 2 seconds
- **API Response Time:** 95% of calls within 200ms
- **Incident Response:** Critical issues addressed within 10 minutes

3. SLAs (Service Level Agreements) outline customer commitments:

- **Availability:** 99.95% uptime commitment
- **Critical Issue Resolution:** Within 60 minutes during business hours
- **Checkout Failures:** Less than 0.5% failed transactions

By aligning SLIs, SLOs, and SLAs, businesses can optimize performance, enhance reliability, and build customer trust.

SRE Solution

1. Observability: Building the Foundation of Reliability

We implemented a robust monitoring framework, leveraging Grafana to track KPIs such as:

- Incoming Request Count – Understanding traffic patterns

- RDS Utilization – Optimizing database performance
- Container CPU and Memory Usage – Monitoring resource consumption
- Running Task Count – Managing workloads efficiently
- Error Rates – Early detection and resolution of issues

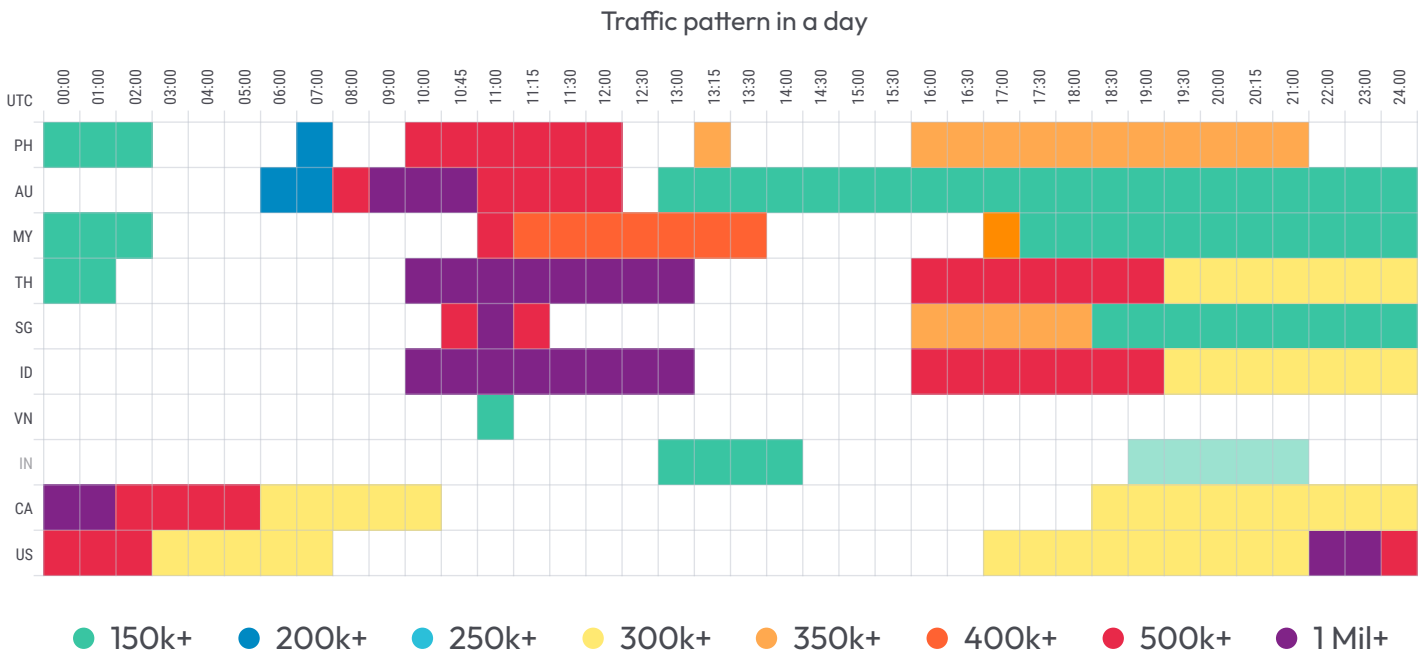
Frequent log scans were conducted to identify critical anomalies, proactively addressing potential failures.

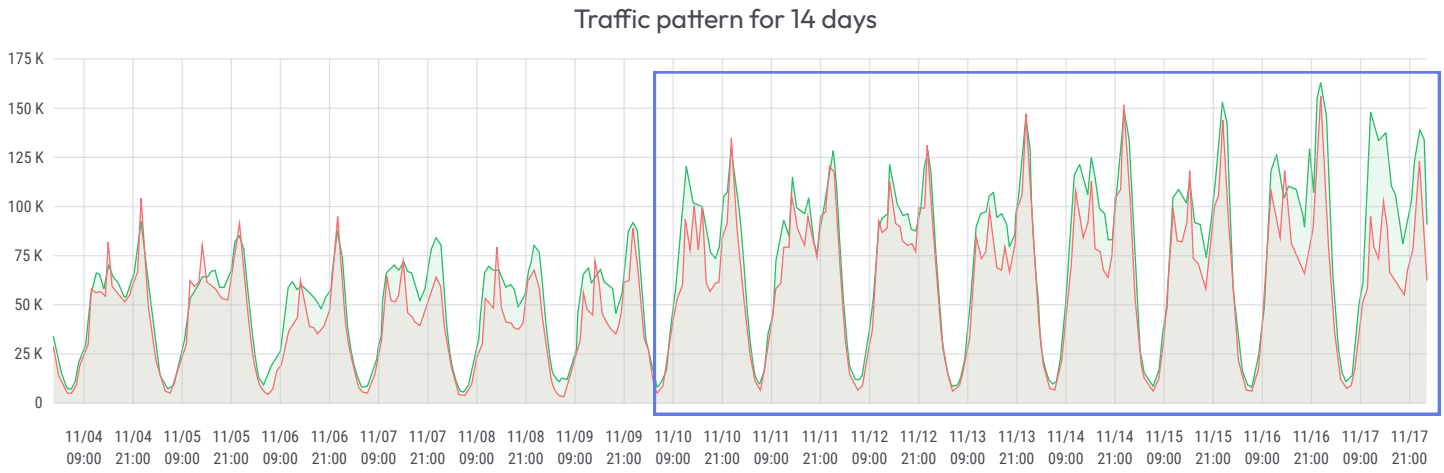
2. Data-Driven Scaling and Cost Efficiency

Through continuous monitoring, we analyzed traffic patterns on a daily, weekly, and monthly basis, which enabled us to:

- Scale down during low-activity hours to reduce costs
- Scale up before peak hours for seamless performance
- Identify peak traffic periods in advance
- Set up automated alerts for traffic exceeding predefined thresholds

This data-driven approach helped us develop a predictive scaling strategy, optimizing costs without compromising performance.





3. Automation: Reducing Manual Effort and Human Errors

We automated infrastructure scaling with Jenkins jobs, enabling scale-ups and scale-downs based on real-time traffic. This resulted in faster responses to traffic fluctuations, reduced overhead, and fewer human errors. We maintain a high maximum container limit, only scaling down the minimum, ensuring quick scaling of the Auto Scaling Group (ASG) during emergencies.

4. Incident Response: Building a Resilient and Prepared Team

A well-prepared team is crucial for minimizing downtime and mitigating failures. Our incident response strategy focused on proactive planning, structured workflows, and ongoing training for swift, effective incident handling.

- Clear Incident Management Framework: We defined a structured flow, ensuring team members knew how to handle system failures, whom to notify, escalation steps, and debugging common failure patterns. We also created detailed manuals with resolution guides, reporting templates, and stakeholder notification protocols.
- Strengthening Preparedness with Mock Drills: Regular mock drills simulated real-world scenarios, reinforcing protocols, identifying gaps, and improving coordination between SRE, DevOps, and support teams.

Through continuous testing and refinement, we improved response times, reduced downtime, and built a confident, prepared team.

5. Root Cause Analysis (RCA): A Learning-Driven Approach

Effective incident management involves staying calm, ensuring recovery within SLA, and conducting thorough RCA.

- No-blame, learning-focused RCA approach that emphasizes learning, not blaming. The goal is to understand what went wrong and prevent recurrence.
- We use the Five Whys method (dig deeper by asking ‘why’ multiple times) to identify the true root cause and ensure corrective actions to target the real problem.
- Structured RCA documentation to capture incident details, root causes, corrective actions, and lessons learned to improve future processes.

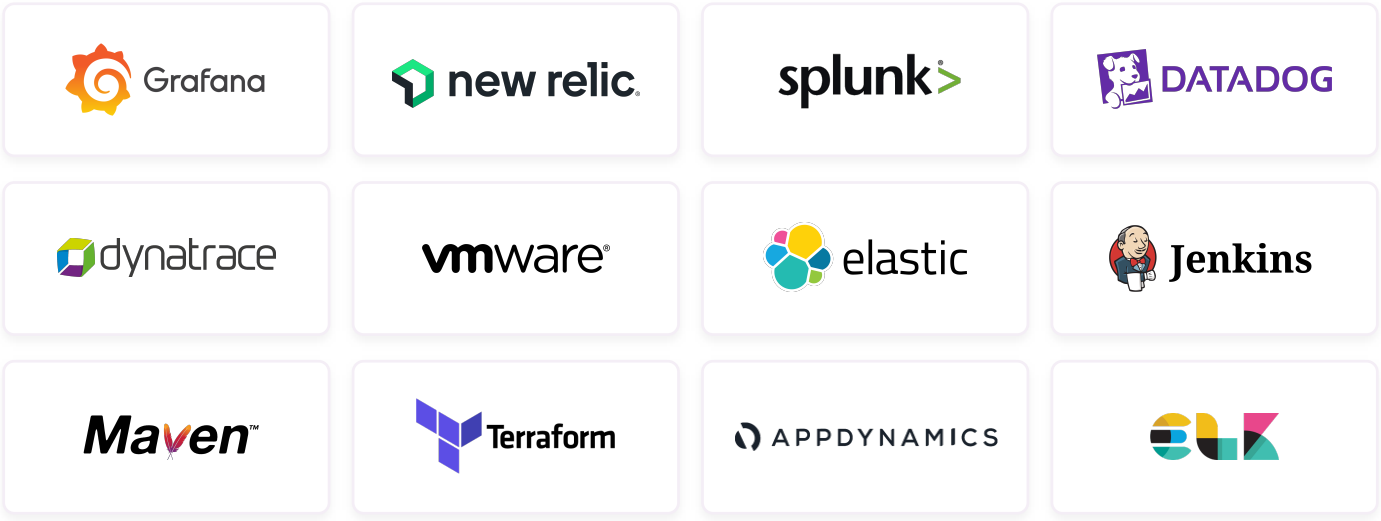
6. Performance Improvements: A Culture of Continuous Optimization

At the core of our SRE approach is a focus on performance optimization, ensuring fast and seamless user experiences.

- Proactive Performance Testing: We conduct regular performance testing with Gatling scripts to identify bottlenecks, assess system response under load, and establish benchmarks for ongoing improvements.
- Query Optimization for Faster Execution: By analyzing slow query logs, we optimized high-latency database queries, improved execution times, refined indexing, and streamlined application logic for better efficiency.

SREs continuously monitor system metrics, identifying and implementing optimizations for enhanced performance.

Technologies



Business Benefits

Cost Savings

- Automated scaling reduced costs by $10K-$13K per region, saving over $1.5 million annually across 10 regions.
- During the Black Friday sale, our approach saved $45K-$50K compared to the previous year.

Proactive Monitoring and Optimization

- Proactive monitoring ensures system stability and cost-effectiveness.
- Data-driven scaling maximizes resource utilization, while automation reduces manual effort and boosts efficiency.

Incident Management and Root Cause Analysis

- A clear Incident Management flow enables quick, organized responses.
- RCA focuses on learning and improvement, preventing recurring issues and enhancing system resilience.

Performance-First Culture

- A performance-driven approach ensures consistent system efficiency and responsiveness through regular testing, early bottleneck detection, and query/logic optimizations.
- A data-driven strategy enables continuous, measurable performance improvements.