RELIABILITY ENGINEERING *in the* CLOUD

Strategies and Practices for AI-Powered Cloud-Based Systems

MARIYA BREYTER | CARLOS ROJAS

FREE SAMPLE CHAPTER |

Praise for *Reliability Engineering in the Cloud*

"As someone deeply invested in achieving operational excellence in the cloud, I found this book to be an absolute game-changer. It's a treasure trove of insights for engineering leaders and software teams eager to boost the reliability of their cloud-based systems. The authors write in a clear, engaging style that makes complex concepts easy to grasp. I was particularly impressed by how they illustrate the role of AI in helping organizations anticipate failures, automate responses, and enhance performance. In today's rapidly evolving digital landscape, I genuinely believe this resource is essential for anyone looking to stay ahead of the curve."

—Valeria Sadovykh, Technology Strategist, Microsoft

"*Reliability Engineering in the Cloud* is a must-read for anyone aiming to build resilient, scalable, and high-performing cloud systems. With actionable insights, real-world case studies, and strategies leveraging cutting-edge technologies, this book offers a comprehensive guide to ensuring system reliability, optimizing operations and fostering a culture of continuous improvement."

—Abhishek Agarwal, author of *Product Mastery*

"Dr. Breyter delivers yet another brilliant work, showcasing her position as a thought leader in enterprise transformation and technical product strategy. Her extensive hands-on experience across diverse industries, mastery of agile methodologies, and commitment to exploring cutting-edge technologies all shine through in this practical book. A must-read for any professional focused on improving service reliability and business results."

—Moshe Rasis, management consultant, executive coach, and faculty at New York University

"This is an essential guide for building resilient, scalable, and fault-tolerant systems. Covering everything from architecture design and incident response to leveraging Gen AI and OKRs, it provides actionable strategies for modern cloud environments. With a focus on automation, observability, and continuous improvement, this book equips teams to master cloud reliability engineering. A must-read for anyone aiming to deliver reliable, high-performing cloud solutions."

—Naveen Ks, Agile coach

"Whether you're a tech leader or a new engineer, this book provides practical guidance on designing resilient architectures and effective incident response using AI, ML, and gen AI to align with OKRs. Mariya Breyter and Carlos Rojas offer an indispensable resource for achieving operational excellence in the AI-cloud era."

—Piyush Sheth, senior lead (Product Enablement, Delivery) at Wells Fargo
Helping Enterprises in their Customer-Centric Product Standup, Enabling Innovation, Delivery Journey

"*Reliability Engineering in the Cloud* by Mariya Breyter and Carlos Rojas masterfully integrates AI-driven analytics and Lean methodologies into cloud reliability engineering. This guide bridges theoretical concepts with practical applications, making it essential for those looking to optimize cloud environments. It's an indispensable resource for leaders and engineers aiming to enhance operational excellence and innovation within their organizations. A must-read for mastering cloud reliability."

—Srinivasaraju Vysyaraju, senior cybersecurity manager

"*Reliability Engineering in the Cloud* is an indispensable guide for technology leaders managing large-scale, distributed cloud systems. Breyter and Rojas deliver actionable insights on AI-driven observability, fault-tolerant architectures, chaos engineering, and operational automation, providing a comprehensive framework to ensure high availability, scalability, and resilience across mission-critical cloud environments."

—Ameesh Paleja, executive vice president, Platform Technology, Capital One

"Knowing that strategies fail without proper execution, I highly recommend this book. It equips engineering leaders to establish lean cloud reliability practices. It offers practical frameworks, crucial training, and leadership development techniques that empower cross-functional teams to drive operational excellence and continuous improvement across complex AI-supported cloud environments."

—Sara Pendergast, president, Advantage by Design, LLC

"An essential guide for enterprise leaders and engineers on how to build reliable and resilient systems to ensure long-term business success in the digital era."

—Daria Kirilenko, former senior director, Information Risk Research at Gartner

"This book is a must-read for anyone building or leading cloud-based systems. Carlos Rojas, a successful technology executive, provides a comprehensive guide to Cloud Reliability Engineering, offering practical strategies, real-world examples, and cutting-edge technologies to ensure resilient, scalable, and reliable applications. A valuable resource for technology leaders!"

—JC Gutierrez, managing director, Technology and Innovation, AWS

Reliability Engineering in the Cloud

Strategies and Practices for AI-Powered Cloud-Based Systems

Mariya Breyter
Carlos Rojas

Addison-Wesley
Hoboken, New Jersey

Cover image: Omelchenko/Shutterstock

Visit us on the Web: informit.com/aw

Library of Congress Control Number: 2025931218

ISBN-13: 978-0-13-539579-0
ISBN-10: 0-13-539579-8

$PrintCode

To my incredible husband, Grigoriy, who for over 30 years has been my rock, my sounding board, and my greatest inspiration—thank you for always believing in me, even when I doubted myself. Max, your success as an aerospace engineer makes me proud every day, and Anthony, your pursuit of excellence as a Presidential Scholarship student gives me hope for the positive impact of your generation on our future. You both are, in your own ways, making this world a better place.

To my extraordinary colleagues over the years: Your diversity of thought has fueled innovative solutions that continuously redefine the boundaries of what's possible in engineering and technology. To all the women in engineering and the students I mentor, thank you for your courage and vision—it's an honor to learn from you and see how you challenge the status quo. The cloud may be a powerful tool, but it's your creativity and perseverance that truly reshape our world.

—Mariya Breyter

To my entire family, especially my dad, who gave me the gift of education and taught me how to be a better man every day. To my kids and wife, who created time for me to focus on this project. To those who offered me words of encouragement during my early days, who trusted me and challenged me to think differently, and who shared their wisdom to build moments that matter throughout my life. You all have been the inspiration to work hard, learn every day, and set time aside to build the next generation of technologists. You all have shown me how important it is to elevate others, how I need to hold the elevator door for the next engineer who is going to be promoted or needs to be recognized. I came to understand early on that their success isn't just their own—it's a shared success that advances us all in society.

—Carlos Rojas

This page intentionally left blank

Contents

Preface

Reliability Engineering in the Cloud Is a Must: Why Your Business Can't Succeed without It

We'd like to start this book with a puzzle. Take a look at these seemingly different catastrophes and their corresponding incident summaries.[1]

- **Streamline Solutions**, a promising tech start-up specializing in e-commerce solutions, experienced rapid growth that brought unforeseen challenges. A sudden surge in website traffic during a major sales event led to a critical outage, resulting in significant revenue loss and damage to the brand's reputation.
  - What happened: Critical website outage during major sales event, lasting six hours
  - Impact: Revenue loss of $100,000, damage to brand reputation resulting in a 20% decrease in customer trust
- **HealthHub Technologies**, a healthcare start-up revolutionizing patient care with telemedicine solutions, encountered operational disruptions when a cloud service outage disrupted critical communication channels between healthcare providers and patients.

1. These and some other use cases in this book are fictional, though based on real companies and actual incidents. When there are actual companies mentioned, it is stated in the text and is referenced in source materials.

    - What happened: Cloud service outage disrupting critical communication channels for 12 hours
    - Impact: A 30% decrease in patient satisfaction, leading to 15% of patients seeking alternative telemedicine providers
- **EcoSolutions**, an environmental start-up dedicated to sustainable energy solutions, encountered operational challenges when a power outage disrupted data collection from IoT sensors deployed across renewable energy installations.
    - What happened: Power outage disrupting data collection from IoT sensors for eight hours
    - Impact: A 50% decrease in energy efficiency, leading to a loss of $50,000 in revenue due to inefficient operations
- **MegaBank Corporation**, a global financial institution, faced a public relations crisis when a technical glitch caused transaction processing delays and disrupted customer access to online banking services.
    - What happened: Technical glitch causing transaction processing delays for 24 hours
    - Impact: A 40% decrease in customer satisfaction, resulting in 10% of customers switching to competitor banks
- **AeroTech Aerospace**, a leading aerospace manufacturer, faced supply chain disruptions when a supplier's data center outage halted production operations.
    - What happened: Supplier's data center outage halted production operations for two days
    - Impact: A $1 million loss in revenue due to halted production, resulting in a 30% decrease in investor confidence
- **MedTech Solutions**, a leading healthcare technology company, faced regulatory compliance challenges when a software bug caused data integrity issues and jeopardized patient safety.
    - What happened: Software bug jeopardized patient safety, affecting the data integrity of 100,000 patient records
    - Impact: Legal fees and settlements totaling $3 million, and a 15% decrease in market share due to damaged reputation
- **Edukate LLC**, an educational technology start-up, encountered performance issues when a surge in user activity overwhelmed its cloud-based learning platform during peak exam periods.

- What happened: Scalability challenges during peak exam periods, resulting in platform downtime of four hours
- Impact: A 25% decrease in user engagement, leading to a loss of $50,000 in subscription revenue

As you can see from these examples, each situation cost millions of dollars and impacted each company's reputation and customer base. If you think this is catastrophic, imagine the consequences for organizations such as hospitals or emergency response systems, where people's lives are at stake.

We are sure that you've guessed by now what all these companies have in common: They neglected to invest in the reliability and resilience of their cloud-based systems. We will refer to this as "cloud reliability engineering" (or "CRE") in this book. In each case, the company was forced to take urgent measures to resolve situations as quickly as possible and then decided to implement long-term resilient strategies and reliability measures. So, what did these companies do? Let's review their solutions one by one.

- **Streamline Solutions:** Determined to prevent future incidents, Streamline Solutions pivoted to prioritize reliability engineering. By leveraging AI-powered predictive analytics, the company optimized resource allocation, identified potential bottlenecks, and implemented proactive measures to ensure uninterrupted service during peak demand periods.
- **HealthHub Technologies:** To mitigate similar incidents in the future, HealthHub Technologies embraced a proactive approach to reliability engineering. By integrating AI-powered monitoring and incident response systems, it achieved real-time visibility into system health, preemptively identified potential issues, and orchestrated automated failover mechanisms to ensure uninterrupted service delivery and patient care.
- **EcoSolutions:** EcoSolutions adopted a proactive approach to reliability engineering, integrating AI-powered predictive maintenance solutions. By leveraging machine learning algorithms to analyze sensor data, detect anomalies, and forecast equipment failures, EcoSolutions optimized asset performance, minimized

downtime, and maximized energy efficiency across its infrastructure.

- **MegaBank Corporation:** MegaBank embarked on a comprehensive reliability engineering initiative, leveraging AI-driven analytics to optimize system performance and enhance customer experience. By harnessing machine learning algorithms to analyze transaction patterns, predict capacity requirements, and dynamically scale resources, MegaBank strengthened its cloud infrastructure's resilience, ensuring seamless banking operations and customer satisfaction.
- **AeroTech Aerospace:** AeroTech implemented a comprehensive reliability engineering strategy augmented by AI-driven supply chain analytics. By harnessing predictive modeling and machine learning algorithms to assess supplier risk, forecast demand fluctuations, and optimize inventory management, AeroTech mitigated supply chain disruptions, ensured uninterrupted production, and maintained customer satisfaction.
- **MedTech Solutions:** MedTech invested in AI-driven quality assurance and compliance solutions. By leveraging machine learning algorithms to automate code analysis, detect potential vulnerabilities, and enforce coding standards, MedTech ensured adherence to regulatory requirements, mitigated compliance risks, and upheld its commitment to patient-centric innovation.
- **Edukate LLC:** Edukate leveraged AI-driven load balancing and auto-scaling mechanisms. By dynamically allocating resources based on user demand, optimizing application performance, and automating capacity provisioning, Edukate enhanced platform reliability, supported growing user engagement, and empowered educators and students with uninterrupted access to educational resources.

It is common knowledge that the cloud is significantly more reliable than on-premises infrastructure. This may sound easy and straightforward, but in real life, it is not. Let's take a closer look at the implementation details for Edukate's AI-driven load balancing and auto-scaling mechanisms, including specific metrics and data.

- **AI-driven load balancing**
  - Edukate's AI-driven load balancer continuously monitors user traffic and application performance metrics in real time.
  - Through historical data analysis, it predicts future user demand patterns with an accuracy rate of over 90%.
  - During peak usage periods, such as final-exam weeks, the load balancer dynamically distributes incoming user requests across multiple servers, ensuring optimal resource utilization.
  - This dynamic load balancing reduces server response time by up to 50% and prevents server overload, maintaining consistent performance levels even under heavy load.
- **Auto-scaling mechanisms**
  - Edukate's auto-scaling mechanisms automatically adjust the number of server instances based on current workload and resource utilization metrics.
  - When user demand increases, the auto-scaler provisions additional server instances within seconds, scaling the infrastructure horizontally to handle the load.
  - During peak traffic hours, the auto-scaler increases server capacity by up to 300%, allowing Edukate to accommodate sudden spikes in user activity without downtime.
  - Conversely, during periods of low demand, excess server instances are automatically terminated, reducing infrastructure costs by up to 40% while maintaining performance.
- **Dynamic resource allocation**
  - Edukate's AI-driven system dynamically allocates resources to different components of the application based on workload and performance requirements.
  - Resources are provisioned and de-provisioned in real time, optimizing CPU, memory, and storage utilization.
  - During peak usage, CPU utilization is maintained below 70% to ensure responsiveness, while memory allocation is adjusted dynamically to prevent bottlenecks.
  - By optimizing resource allocation, Edukate achieves an average server utilization rate of 80%, significantly reducing infrastructure waste and costs.

- **Optimizing application performance**
  - Edukate continuously monitors application performance metrics using AI-driven monitoring tools, including response time, throughput, and error rates.
  - Through real-time analysis, performance bottlenecks are identified and addressed promptly, resulting in a 40% improvement in application response time.
  - Database optimizations, such as query caching and indexing, reduce database response time by up to 60%, enhancing overall application performance.
  - Network optimizations, including content delivery network (CDN) integration, reduce latency by 30%, resulting in faster content delivery to users worldwide.

Overall, Edukate's implementation of AI-driven load balancing and auto-scaling mechanisms has led to significant improvements in performance, scalability, and cost efficiency. By leveraging predictive analytics and automation, Edukate ensures uninterrupted access to its cloud-based learning platform for educators and students, even during periods of peak demand.

In today's digital age, where every click and keystroke carries immense significance, the impact of CRE is hard to overstate. With the majority of systems, both internal and external, now residing in the cloud, reliability has emerged as the foundation of a successful business. From catastrophic events that have compromised millions of people to crippling service outages eroding customer trust and costing companies millions of dollars in revenue—the stakes have never been higher. With the huge repercussions of engineering misses, strong resilience engineering practices play a pivotal role in the modern business landscape and can elevate a business or destroy it. With the ability to leverage AI in multiple areas of CRE, including load balancing and auto-scaling, monitoring and alerting, risk assessment, anomaly detection, and issue resolution, CRE practices can make or break any business. As we saw in the examples, companies big and small in every possible industry can't survive without a strong CRE strategy and an AI-powered implementation of this strategy.

But first, let's start with the basics. While our primary focus in this book will be on AI-powered CRE, our secondary focus will be on efficiency, which can be achieved with Lean CRE. Lean CRE encompasses an array of principles and practices, borrowing from the traditions of Toyota manufacturing and Lean Six Sigma methodologies (see Figure I.1). These proven approaches have redefined efficiency and effectiveness in manufacturing and process optimization. Now, companies big and small are innovating in the field of CRE by adding a focus on software and people investments, and for compelling reasons. Mastering these techniques is non-negotiable for modern businesses. Successful CRE practice needs to be Lean to successfully prevent colossal failures that have happened within some of the world's most prominent companies, and to enable structured and well-thought-out CRE practices and CRE culture.

Figure I.1
Six Sigma pillars
(image: Trueffelpix/Shutterstock)

As we dive deep into the topics of system reliability and resilience, it's crucial to understand the fundamental distinction between site reliability engineering (SRE) and CRE. While SRE primarily focuses on maintaining the reliability of individual systems and services, CRE takes a more holistic approach, encompassing the entire digital infrastructure: the cloud services used for the control plane and data plane, the tooling to monitor and observe, the AI practices to predict failure, and the mechanisms to stress-test systems using chaos engineering practices, ensuring that the entire ecosystem operates harmoniously and efficiently—with operational excellence. CRE, in this context, becomes the linchpin that holds together an organization's digital success, while AI technologies provide a strong foundation for its implementation.

CRE is not just a choice but an imperative for those organizations that aim to succeed in the digital realm. This book will show engineering leaders and decision-makers how AI-powered implementation and Lean principles can revolutionize the world of CRE and why they should be at the forefront of your organization's strategy, and offer practical advice on implementing effective reliability practices, selecting the right tools and frameworks, measuring impact through meaningful metrics, and building organizational culture that ensures sustained success in a cloud-driven ecosystem.

To summarize, reliability engineering in the cloud fundamentally changes the way organizations—big and small, profit and nonprofit, in any industry and any country—operate their systems and engineering teams. Given the rapidly evolving landscape of cloud technology, the principles of reliability engineering are undergoing a profound transformation, propelled by the integration of AI. In this book, we will take you on a transformative journey from the initial steps in building and maintaining reliable and resilient infrastructure and applications for your business, to nurturing CRE culture across the company. We will discuss the best strategies to address recovery, dissect resilience challenges, implement AI-driven frameworks, and provide the foundation that will fortify your business's digital infrastructure. It is time for your company to rewrite your engineering playbook using the methodologies and best practices described in this book. Let the journey begin.

Who Is This Book For?

This book will benefit organizations responsible for building systems in the cloud, and those engineering leaders who need to set an enterprise-wide strategy with thousands of applications and dependencies. Think of it like a corporation with hundreds or thousands of dev teams. This book serves as both the framework and a reference to two groups of readers.

The first group consists of enterprise leaders, from director and VP level to heads of specific businesses, who are looking to increase the reliability and scalability of their systems in the cloud, the efficiency of their operations, and their need for faster incident response while automating their operations to improve time to restore and time to detect to the maximum possible extent. These leaders know that operational agility and chaos experimentation bring a culture of continuous improvement built on collaboration and knowledge sharing among teams.

The second group includes engineers and software teams directly involved in or responsible for cloud applications' reliability. The frameworks and guides described in the book will allow them to build effective strategies, promoting chaos engineering practices, observability and monitoring techniques, disaster recovery exercises, reliability metrics, fast data-driven decision-making, and practical examples of techniques and tooling for success. Given the lack of both literature on the topic and established frameworks, this group of readers will benefit from having practical, domain-specific approaches and examples that they can apply to their organizations and teams.

For both groups, adopting modern CRE practices will better position their organizations to build and maintain resilient systems that meet their customers' demands to achieve their business goals effectively. This book focuses on examples and techniques used with cloud providers such as AWS, GCP, and Azure.

Acknowledgments

To my colleagues at some of the world's most innovative tech and financial services companies, your insights, dedication, and collaboration have been invaluable in shaping this book. And to my customers, your challenges have fueled my customer obsession and driven me to dig deeper into what it means to be truly reliable in the cloud.

To every reader who picks up this book, my hope is that it inspires you to act boldly and make our digital world more secure and its information accessible to all. A special thank-you to my coauthor, Carlos Rojas—your leadership continued to pave the way for transformation at a leading cloud provider and now at a global financial services company. Your vision for positive change is nothing short of inspiring.

Here's to the next generation of innovators, the ones who will push the boundaries of cloud reliability even further.

—Mariya Breyter

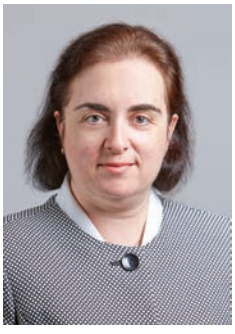
To my peers, leaders, teams, and friends who worked with me to transform companies during my career: For the trying moments when we collectively learned something new, I am confident most of those became components shared in this book. The concepts of reliability engineering started way before we even had the cloud. Our learnings determined how today the most important companies run their critical services in the cloud.

To every reader who decides to invest in CRE, my goal is that you implement at least one bold idea from this book to make your organization more reliable. This book will open the world of ideas to continue to invest in the highest engineering standards in recent times.

A special thank-you to my coauthor, Mariya Breyter—your leadership while we worked at AWS and your friendship outside of work have no limits and inspired me to complete this amazing project. Your vision of continued improvement and dedication to be the best is simply contagious.

—Carlos Rojas

About the Authors



Mariya Breyter is a technology executive, product leader, and educator with experience ranging from leading technology companies and government jobs to versatile corporate experience in cloud services, healthcare, financial services, media, and education. She takes pride in leading high-performance organizations in building technology products that delight customers, whether it is cloud services at a leading cloud provider, the online consumer bank Marcus by Goldman Sachs, an educational start-up incubator at Kaplan Test Prep, or a simplified global claims system at UnitedHealth Group. Her book *Agile Project and Product Management*, published in 2022, was added to Amazon's list of the top 100 books on Agile and to Book Authority's list of the "Best Agile Software Development Books of All Time," and her article on Agile frameworks was selected as part of the top Agile Articles 2020. Mariya has a PhD in computational linguistics and a post-doctorate degree from Stanford University. Mariya also teaches Agile project management, IT principles, and organizational transformation courses to graduate students at New York University. Her passion is to help others achieve their professional growth. She founded Agile Practitioners Meetup in New York, organized sessions for Women in Agile/New York, and led Mentoring Circles at the Grace Hopper Celebration. Mariya's motto is, "Lead with passion, guide with purpose, inspire with innovation."

Connect with Mariya: https://www.linkedin.com/in/mariyabreyter/

Carlos Rojas is a former Amazon Web Services (AWS) engineering leader; is a technology advisor with a certification in AI from Massachusetts Institute of Technology; and has 25+ years of experience in application development, reliability engineering, operational excellence, and IT shared services, with a proven record of success within financial, telecom, government, and health IT. Carlos is well known across multiple industries for building and transforming teams. Carlos began working as a freelance software developer in 1995 when the internet was looming. Following that, for a decade he fulfilled his entrepreneurial drive with a tech start-up called Cascade Technologies. Over the past several years, Carlos used his skills to build and transform teams into forward-thinking industry leaders in multiple leadership roles. As the global head of Region Build Automation at AWS, Carlos was part of the engineering leadership team responsible for orchestrating the launch of new regions across the globe. In 2022 he came back to a major bank as the vice president of Cloud Reliability Engineering, responsible for reimagining customers' digital experiences and improving the operational excellence of their applications.

Connect with Carlos: https://www.linkedin.com/in/carojas77/.

The combined in-depth experiences in CRE, Lean, cloud services, and enterprise engineering allow the authors to view enterprise needs from multiple perspectives. Their prior experience at AWS gives them a joint perspective on successfully applying these practices at a significant scale, and their network of content providers with specific domain expertise makes this a must-have book.

Chapter 7
CRE Tooling
Tools That Support Automatic Failovers, Automatic Rollbacks, Automatic Deployments, Chaos Engineering, Incident Response, Configuration Management, Immutable Infrastructure, and Disaster Recovery

Proper tooling is essential in cloud reliability engineering (CRE) to maintain the reliability, availability, and performance of cloud-based systems. Automation helps streamline recovery operations, reduce manual intervention for testing scenarios, and ensure that teams can proactively respond to issues. Cloud providers offer a large number of automation tools that embrace these principles and techniques. In this chapter, we will review some of the tools and discuss why they are important in CRE.

Distributing Load and Volume with Auto-Scaling and Load Balancing

Reliability engineering focuses on measuring how resilient, stable, and scalable your systems are. This requires distributing and balancing loads to ensure an "always on" posture for your most critical systems. Amazon Web Services' (AWS) Well-Architected Tool is an example of a tool that allows you to conduct reviews of your applications according to AWS architectural best practices. It provides a structured framework for assessing your architecture, identifying areas in need of improvement, and making informed decisions to optimize your AWS workloads. Let's take a look at how application teams can use this tool to configure and test the resilience, stability, and scalability of their systems.

Auto-Scaling

Cloud auto-scaling is a cloud computing feature that automatically adjusts the number of compute resources (e.g., virtual machines [VMs]) allocated to an application based on changing demand. The primary goal of auto-scaling is to ensure that applications can handle varying levels of traffic and workloads efficiently and without manual intervention. This is where the concept of elasticity becomes a major player in your cloud implementations.

AWS Auto Scaling (see Figure 7.1) automatically adjusts the number of instances in response to changes in demand, ensuring that applications are neither overprovisioned nor underprovisioned. An example of auto-scaling is the dynamic resource allocation that occurs when AWS Auto Scaling monitors the performance and resource utilization of your application. When certain predefined conditions are met, such as increased traffic or CPU utilization, AWS Auto Scaling automatically provisions additional resources based on scaling policies, such as CPU usage metrics, network traffic, or custom application-specific metrics. When demand decreases, it can scale down resources to avoid overprovisioning and reduce costs.

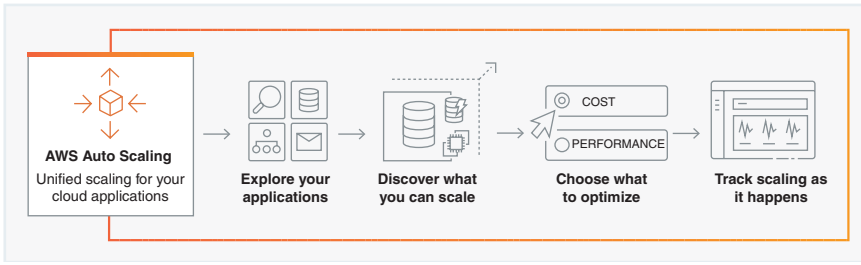


Figure 7.1
AWS Auto Scaling (source: https://aws.amazon.com/autoscaling/; © 2024, Amazon Web Services, Inc.)

Auto-scaling also provides elasticity to your applications, allowing your systems to seamlessly handle traffic spikes and other fluctuations in demand without manual intervention. This elasticity contributes to high availability and improved performance.

Finally, by automatically scaling resources up and down, auto-scaling helps optimize cloud costs so that you only pay for the resources you use, which can lead to cost savings during periods of lower demand. AWS provides multiple resources for cost optimization, including AWS Cost Optimizer, AWS Cost Explorer, and AWS Cost Estimator.

GCP provides the following auto-scaling tools.

- **Google Kubernetes Engine (GKE) Autoscaler:** GKE Autoscaler automatically adjusts the number of nodes in a given cluster based on the demands of your workloads. It can scale nodes based on various metrics, including CPU utilization and custom metrics.
- **Google Compute Engine Autoscaler:** This tool adjusts the number of instances in a managed instance group based on the current load. It can scale based on CPU utilization, HTTP load balancing, server capacity, and custom metrics.
- **Google App Engine Autoscaling:** Google App Engine provides automatic scaling based on request rates, response latencies, and other application metrics. It ensures that your application always has enough instances to handle incoming traffic.

Microsoft Azure offers the following options:

- **Azure Autoscale:** Azure Autoscale enables you to automatically adjust the number of compute resources based on demand. It supports scaling based on metrics such as CPU usage, queue length, and schedule-based scaling.
- **Azure Virtual Machine Scale Sets:** Azure Virtual Machine Scale Sets allows you to create and manage a group of identical load-balanced VMs. You can define auto-scale rules based on CPU usage or other metrics to automatically adjust the number of VM instances.
- **Azure App Service Autoscale:** Azure App Service Autoscale offers auto-scaling capabilities for web apps, API apps, and mobile apps. It can scale instances horizontally based on metrics such as CPU usage, memory usage, and HTTP queue length.

Figure 7.2 illustrates how auto-scaling can be configured to create the most efficient and reliable posture for your applications. The figure depicts how the number of VMs will remain at two when the application experiences minimum volumes; as new workloads and users connect, the infrastructure will be elastic to support the additional load and grow from two to a maximum of five VMs. In this scenario, the application has found the need to increase to three VMs based on a condition, whether it be CPU usage, memory usage, or HTTP queues.

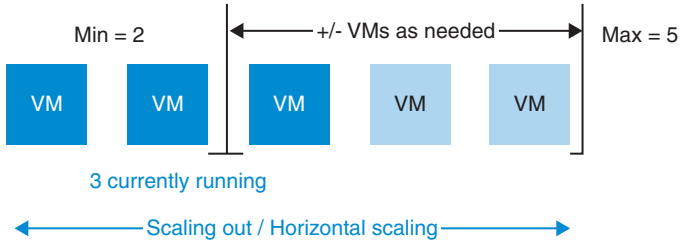


Figure 7.2
Azure Autoscale (source: https://learn.microsoft.com/en-us/azure/azure-monitor/Autoscale/autoscale-overview/; © 2024, Microsoft)

Load Balancing

Load balancing is a technique used to distribute incoming network traffic or requests across multiple servers or computing resources. The primary purpose of load balancing is to ensure that no single server or resource is overwhelmed by traffic, thereby improving the availability, fault tolerance, and performance of applications. AWS Elastic Load Balancing (ELB) distributes incoming application traffic across multiple targets, increasing availability and fault tolerance.

Load balancing includes traffic distribution so that load balancers evenly distribute incoming requests or network traffic across a pool of resources, ensuring efficient resource utilization. Also, load balancers monitor the health and status of services, and reroute traffic (if a server becomes nonresponsive) to ensure high availability by distributing traffic across multiple regions, thereby improving application resilience and performance.

All major cloud providers offer load-balancing tools. Following is a sample of those available.

- **Google Cloud Load Balancing:** Google Cloud Load Balancing distributes incoming traffic across multiple instances or backend services to ensure high availability and reliability of your applications. It offers several types of load balancers:
  - **HTTP(S) load balancing:** For distributing HTTP and HTTPS traffic across multiple backend instances or services
  - **TCP proxy load balancing:** For distributing TCP traffic to backend instances
  - **SSL proxy load balancing:** For distributing SSL/TLS traffic to backend instances
  - **Internal TCP/UDP load balancing:** For distributing internal TCP and UDP traffic within your virtual private cloud (VPC) network
- **Azure Load Balancer:** Azure Load Balancer distributes incoming network traffic across multiple VM instances in a backend pool. It supports both inbound and outbound scenarios and can be configured for various protocols including TCP, UDP, and HTTP/S.

- **Azure Application Gateway:** Azure Application Gateway is a layer 7 load balancer that provides application-level routing and load-balancing services. It offers features such as SSL termination, cookie-based session affinity, URL-based routing, and web application firewall (WAF) capabilities.
- **Azure Traffic Manager:** Azure Traffic Manager is a domain name system (DNS)–based traffic load balancer that distributes incoming traffic across multiple endpoints located in different Azure regions or globally. It provides various load-balancing methods including priority, weighted, performance, and geographic routing.

Table 7.1 outlines Azure's load-balancing methods and features.

Table 7.1
Azure's Load-Balancing Methods and Features

	Azure Traffic Manager	Azure Application Gateway	Azure Front Door	Azure Load Balancer
OSI layer	7	7	7	4
Health probes	HTTP/HTTPS/TCP	HTTP/HTTPS	HTTP/HTTPS	TCP/HTTP
SKUs	—	Basic/standard	—	Basic/standard
Load balancing	Global	Regional	Global	Global
Works at:	VMs	Any IP address	DNS CNAME	—
TCP and UDP	DNS	HTTP/HTTPS/HTTP2/WS	HTTP/HTTPS/ HTTP2	TCP and UDP
Sticky sessions	Supported	Supported	Supported	Supported
Traffic control	—	Network Security Group	—	Network Security Group
WAF	—	WAF	WAF	—

All of these load-balancing tools and features help distribute incoming traffic across multiple backend instances or services to ensure high availability, scalability, and performance of your applications.

With sticky sessions, a load balancer assigns an identifying attribute to a user by issuing a cookie or by tracking the user's IP details. Then, according to the tracking ID, the load balancer can start routing all the user's requests to a specific server for the duration of the session. This creates a seamless and stable experience for users, as they will get latency responses similar to those they would get if they were receiving service from hosts and apps within the same load balancer perimeter.

Cloud auto-scaling and load balancing are fundamental techniques for ensuring that your applications can efficiently handle varying workloads, maintain high availability, and optimize resource utilization in the cloud. Auto-scaling adapts to changing demand by adjusting the number of resources, while load balancing evenly distributes traffic to prevent overload and improve fault tolerance. Together, these technologies help create robust and responsive cloud-based applications.

Enabling Automatic Failovers for High Availability

Enabling automatic failovers for high availability is a critical aspect of cloud infrastructure design. It ensures that your applications and services remain accessible and operational, even in the face of hardware failures, network issues, or other unexpected events. The following services play a vital role in enabling automatic failovers for high availability in your cloud-based applications. Depending on your specific use cases and requirements, you can leverage one or more of these tools to design a resilient and fault-tolerant infrastructure that ensures continuous availability and minimal downtime for your applications and services.

AWS

AWS provides several tools and services that enable automatic failovers to achieve high availability.

- **Amazon Route 53 DNS Failover:** Route 53 DNS Failover is AWS's scalable DNS web service. It offers DNS failover capabilities, allowing you to automatically reroute traffic from an unhealthy or unavailable resource to a healthy one based on health checks. In essence, you can use Route 53 DNS Failover to ensure high availability of your web applications, websites, and services by directing traffic to healthy endpoints in the event of failures.
- **Amazon Relational Database Service (RDS):** RDS is a managed database service that offers multi-AZ deployments for database instances. Multi-AZ provides high availability by replicating your primary database to a standby instance in a different Availability Zone (AZ). By using multi-AZ, you can ensure that your database remains available with automatic failover in the event of a database instance failure or planned maintenance.
- **Amazon RDS Aurora:** RDS Aurora is a highly available and scalable relational database service. Aurora Multi-Master allows you to create multiple read/write master database instances for high availability and read scalability. Aurora Multi-Master is suitable for applications that require both high availability and the ability to distribute write workloads across multiple database instances.
- **Amazon DynamoDB:** DynamoDB is a managed NoSQL database service. DynamoDB global tables enable you to create multiregion, multi-active databases to provide high availability and low-latency access to your data. Global tables are ideal for global applications that need to maintain high availability across multiple geographic regions.

- **AWS Global Accelerator:** Global Accelerator is a network service that provides high availability and fault tolerance for applications deployed across multiple AWS regions. Global Accelerator helps route traffic to healthy endpoints in the event of failures or performance degradation, improving the availability and responsiveness of your application.
- **Amazon Simple Storage Service (S3):** S3 offers data replication options, including cross-region replication (CRR) and same-region replication (SRR). These options enable data replication for high availability and data durability. S3 replication is crucial for ensuring that your data remains accessible and intact, even in the face of region-specific failures or disasters.

GCP

GCP provides the following tools that enable automatic failovers to achieve high availability.

- **Google Cloud Load Balancing:** Google Cloud Load Balancing provides built-in failover capabilities to ensure high availability of your applications. It continuously monitors the health of backend instances or services and automatically redirects traffic away from failed instances to healthy ones. This helps minimize downtime and ensures that your applications remain accessible, even in the event of failures.
- **Google Cloud DNS:** Google Cloud DNS offers automatic failover functionality for DNS records. You can configure DNS failover policies to monitor the availability of your backend services and automatically update DNS records to redirect traffic to alternative IP addresses or endpoints in case of failures. This helps ensure seamless failover and continuous availability of your applications.

Microsoft Azure

Microsoft Azure provides the following tools that enable automatic failovers to achieve high availability.

- **Azure Traffic Manager:** Azure Traffic Manager provides automatic failover capabilities for distributing traffic across multiple endpoints located in different Azure regions or globally. It continuously monitors the health of endpoints and automatically redirects traffic away from failed endpoints to healthy ones. This helps ensure high availability and reliability of your applications by minimizing downtime and maintaining continuous access for users.
- **Azure App Service:** Azure App Service offers built-in autohealing capabilities for web applications hosted on the platform. It automatically detects and resolves common application issues, such as crashes or unresponsiveness, by recycling or restarting the affected instances. This helps minimize downtime and ensures that your web applications remain available and responsive to users.

Facilitating Controlled Deployments with Rollback Strategies

Facilitating controlled deployments with rollback "n–1 stack" strategies is a software deployment approach in which, during a software update or release, a new version of software is deployed to all but one of the available environments. This one environment is typically referred to as the "n–1" environment, meaning it represents the previous version of the software.

The purpose of this strategy is to maintain a fallback option in case any critical issues or unexpected problems arise with the new software release. If issues are detected in the newly deployed version,

the organization can quickly switch back to the n–1 version, minimizing downtime and potential disruptions.

To facilitate controlled deployments, cloud providers offer several tools.

- **AWS CodeDeploy** is a fully managed deployment service provided by AWS. It automates the deployment of applications to various compute services, including Amazon Elastic Compute Cloud (EC2) instances, AWS lambda functions, and on-premises servers. CodeDeploy makes it easier to release new features, updates, and bug fixes while ensuring that deployment processes are consistent and reliable. CodeDeploy allows you to define deployment configurations, rollbacks, and monitoring options. It supports various deployment strategies, including n–1 deployments, blue-green deployments, and canary deployments.
- **AWS CodePipeline** is a fully managed continuous integration and continuous delivery (CI/CD) service that automates the building, testing, and deployment of applications. It allows developers to define and automate their release processes, from source code changes to production deployments, using customizable pipelines. CodePipeline supports integrations with various AWS services and third-party tools, making it a versatile solution for streamlining the software delivery lifecycle.
- **AWS Elastic Beanstalk** is a platform as a service (PaaS) offering that simplifies the deployment and management of applications. Developers can easily deploy web applications and services written in various programming languages, such as Java, Python, Node.js, and more, without dealing with the underlying infrastructure details. Elastic Beanstalk provides automated scaling, load balancing, and monitoring, allowing developers to focus on writing code while AWS handles the deployment and scaling aspects of their applications.

The combined value of AWS CodeDeploy, AWS CodePipeline, and AWS Elastic Beanstalk lies in their ability to automate and

streamline the entire application development and deployment process. CodePipeline orchestrates the CI/CD pipeline, enabling efficient code changes from development to production. CodeDeploy automates application deployments, ensuring consistency and reliability, while Elastic Beanstalk simplifies application management, allowing engineers to focus on code rather than infrastructure. Together, these services promote a Lean approach to CRE by reducing manual intervention, enhancing deployment efficiency, and optimizing resource utilization, ultimately improving the reliability and resilience of cloud-based applications.

Google and Azure options include the following.

- **Google Cloud Deployment Manager:** Google Cloud Deployment Manager is an infrastructure as code (IaC) service that allows you to define and manage your cloud resources using declarative configuration files. You can define the desired state of your infrastructure in configuration files written in YAML or Jinja2 templates, and Deployment Manager will automatically create, update, or delete resources to match the desired state. This enables you to manage your deployments in a controlled and repeatable manner, with the ability to roll back changes if needed.
- **Google Kubernetes Engine:** GKE is a managed Kubernetes service that allows you to deploy, manage, and scale containerized applications using Kubernetes. Kubernetes provides built-in features for controlled deployments, such as rolling updates and canary deployments. With GKE, you can define deployment strategies, such as blue-green deployments or rolling updates, to control the rollout of new application versions.
- **Azure Resource Manager (ARM):** ARM is the infrastructure deployment and management service in Azure that allows you to provision and manage your cloud resources using templates. ARM templates are JSON files that define the desired state of your infrastructure, including VMs, storage accounts, networking resources, and more. You can use ARM templates to create, update, or delete resources in a controlled and repeatable manner, enabling consistent deployments across environments.

- **Azure DevOps:** Azure DevOps is a suite of cloud-based collaboration tools for software development, including version control, build automation, release management, and more. Azure DevOps provides features for controlling deployments, such as release pipelines, deployment gates, and approvals. You can define release pipelines that automate the deployment process, and include gates or approval steps to control when and how changes are deployed to different environments such as development, testing, staging, and production.

Providing Chaos Engineering Capabilities for Resilience Testing

Chaos engineering is a crucial practice in modern cloud and DevOps environments. Cloud providers developed several tools that offer chaos engineering capabilities for resilience testing, helping organizations proactively identify and address weaknesses in their systems. Some of these tools include the following.

- **AWS Fault Injection Simulator (FIS):** FIS allows you to run controlled chaos experiments on your infrastructure to test its resilience. You can introduce faults and failures into your AWS resources to see how your systems respond. FIS supports a variety of AWS services and failure modes, making it a powerful tool for assessing your application's reliability.
- **AWS Systems Manager:** While Systems Manager is primarily used for managing and automating operational tasks, it also includes features for running maintenance and compliance tasks, which can simulate failures and test the resilience of your systems. It offers a broader set of capabilities beyond chaos engineering, but it can be leveraged for such purposes.
- **AWS Step Functions:** Step Functions can be used to design and execute workflows that simulate failure scenarios and test how your applications react.

- **Chaos Mesh:** Chaos Mesh is an open source chaos engineering platform for Kubernetes environments, developed by the Cloud Native Computing Foundation (CNCF) community. It allows you to inject faults and disturbances into your Kubernetes clusters to simulate real-world failures and test the resilience of your applications and infrastructure. Chaos Mesh supports various chaos engineering experiments, such as Pod failure, network latency, packet loss, and more. You can define chaos experiments using Chaos Mesh's Custom Resource Definition (CRD) API and specify the scope, duration, and severity of the injected faults.
- **Azure Chaos Studio:** Azure Chaos Studio is a chaos engineering service for Azure that allows you to simulate real-world failures and test the resilience of your cloud applications and infrastructure. It provides a user-friendly web-based interface for creating, running, and analyzing chaos experiments. Azure Chaos Studio integrates with Azure Monitor and Azure Resource Manager to discover and target resources in your Azure environment for chaos testing. You can define chaos experiments to inject faults and disturbances, such as network latency, VM failures, service interruptions, and more, and observe the impact on your applications' performance and availability.

Assisting in Incident Response with Automation

Incident response and automation are integral components of CRE, and AWS offers a suite of powerful tools to assist organizations in effectively managing incidents and automating responses.

- **AWS CloudWatch alarms** enable proactive monitoring by allowing you to set alarms on various metrics, triggering automated actions when specific thresholds are breached. This feature empowers teams to respond swiftly to issues and minimize the impact on system reliability.

- **AWS CloudWatch Events** further enhance incident response by providing a real-time stream of system events and changes, which can be used to trigger automated workflows. These events can be integrated with AWS Lambda, a serverless compute service that executes code in response to various events, such as log file uploads or alarms. Lambda functions can be customized to automate incident response actions, enabling organizations to mitigate issues automatically and without manual intervention.
- **AWS Simple Notification Service (SNS)** plays a pivotal role in incident communication and alerting. It allows for the distribution of real-time notifications through various channels such as email, SMS, or HTTP endpoints. During incidents, SNS can be used to leverage application-to-people communications to notify relevant team members and stakeholders, or trigger automated incident resolution processes.
- **AWS Systems Manager** is a comprehensive tool that aids in managing and automating operational tasks across AWS resources. It facilitates the orchestration of incident response activities, such as patch management, configuration compliance, and instance management. By streamlining these tasks, AWS Systems Manager ensures that incidents are handled efficiently and with minimal disruption to system reliability. In essence, this suite of AWS tools empowers organizations to respond to incidents swiftly and automate key aspects of the incident resolution process, enhancing the overall reliability of cloud-based systems.

Google Cloud and Azure offer the following comprehensive monitoring, logging, and diagnostic services that can further enhance the speed and efficiency of incident detection and resolution:

- **Google Cloud Operations Suite:** Google Cloud Operations Suite provides a comprehensive set of monitoring, logging, and diagnostics tools to help you gain insight into the performance, availability, and health of your applications and infrastructure on GCP. It includes features such as Monitoring, Logging, Trace, Debugger, Profiler, and Error Reporting.

With Monitoring, you can set up alerts and notifications to detect and respond to incidents in real time. Logging allows you to centralize and analyze logs from your applications and services. Trace provides distributed tracing for understanding request latency and performance bottlenecks. Debugger allows you to inspect the state of your applications in production. Profiler helps you optimize the performance of your applications. Error Reporting aggregates and analyzes error events to help you diagnose and fix issues quickly.

- **Azure Monitor:** Azure Monitor is a comprehensive monitoring service for Azure that provides insights into the performance, availability, and health of your applications and infrastructure. It includes features such as Metrics, Logs, Alerts, Application Insights, and Azure Automation. With Metrics, you can monitor the performance and health of your Azure resources and set up alerts based on predefined thresholds or custom queries. Logs allows you to collect, analyze, and visualize log data from your applications and services. Alerts enables you to configure alert rules to notify you when specific conditions are met. Application Insights provides application performance monitoring (APM) and application analytics for your Azure and on-premises applications. Azure Automation allows you to automate the response to incidents and events by defining runbooks and workflows that perform remediation actions.

Ensuring Proper Configuration Management

Ensuring proper configuration management and compliance is a critical aspect of CRE, and AWS Config is a robust tool designed to address these needs comprehensively. AWS Config continuously monitors and records configuration changes to AWS resources, providing a detailed history of these modifications. This historical data allows organizations to assess and audit their resource configurations,

helping to identify and rectify discrepancies or potential security vulnerabilities promptly.

AWS Config also plays a vital role in maintaining compliance with regulatory requirements and industry standards. It allows organizations to define and enforce desired configurations through rules and policies, ensuring that their AWS resources adhere to best practices. When any configuration drift occurs, AWS Config can trigger automated remediation actions or send alerts, enabling organizations to maintain a consistent and compliant infrastructure while minimizing manual intervention. Overall, AWS Config provides a robust foundation for configuration management and compliance, helping organizations enhance the reliability and security of their cloud-based environments.

AWS AppConfig is a platform that specializes in configuration management solutions for mobile applications. One public case that demonstrates the value of AWS AppConfig and configuration management best practices is its collaboration with a major mobile banking application.

In this case, the mobile banking application was facing challenges in delivering personalized experiences to its users while ensuring security and compliance with regulatory requirements. The app needed to dynamically adjust its features, user interface elements, and backend services based on factors such as user preferences, device capabilities, and regulatory changes. However, managing these configurations across a large user base and diverse device landscape was becoming increasingly complex and error-prone.

By implementing the AWS AppConfig solution, the mobile banking application was able to streamline the management of its configurations and achieve several key benefits.

- **Dynamic personalization:** AWS AppConfig allowed the mobile banking app to dynamically personalize the user experience based on factors such as user preferences, location, and device type. By centralizing configuration management, the app could easily adjust its features and content without requiring app updates or manual intervention.

- **Enhanced security and compliance:** AWS AppConfig provided robust security features, such as encryption, access controls, and audit logs, to ensure that sensitive configuration data was protected from unauthorized access or tampering. Additionally, AWS AppConfig helped the mobile banking app maintain compliance with regulatory requirements by enabling granular control over configuration changes and versioning.
- **Improved agility and time to market:** With AWS AppConfig, the mobile banking app could quickly iterate on new features, experiment with different configurations, and roll out updates to specific user segments in real time. This agility helped the app stay ahead of competitors and respond rapidly to changing market demands.
- **Reduced operational overhead:** By automating the deployment and management of configurations, AWS AppConfig reduced the operational overhead associated with manual configuration tasks and troubleshooting. This freed up resources for the mobile banking app's development and operations teams to focus on strategic initiatives and innovation.

This collaboration between AWS AppConfig and the mobile banking app showcases the value of configuration management best practices in enabling dynamic personalization, enhancing security and compliance, improving agility, and reducing operational overhead in mobile application development and delivery.

Leveraging Immutable Infrastructure as a Service

Infrastructure as a service (IaaS) is a fundamental building block in cloud computing, and AWS CloudFormation is AWS's premier service for managing and provisioning cloud IaC (see Figure 7.3). AWS CloudFormation allows users to define and provision AWS infrastructure resources using a declarative template, typically in JSON or YAML

format. These templates describe the desired state of the infrastructure, including compute resources, storage, networking, and more, in a human-readable and version-controlled manner.

One of the primary benefits of AWS CloudFormation is the automation it brings to infrastructure management. By codifying infrastructure definitions, organizations can version-control their infrastructure, enabling better collaboration among teams and simplifying resource provisioning and management. This automation reduces the risk of manual configuration errors and streamlines the process of creating, updating, and deleting resources as needed. AWS CloudFormation also supports rolling updates and allows for the efficient scaling of resources, making it a valuable tool for maintaining a reliable and responsive cloud environment. Whether you're launching a single-instance application or managing a complex, multitiered architecture, AWS CloudFormation provides the flexibility and automation needed to ensure the reliability and consistency of your cloud infrastructure.

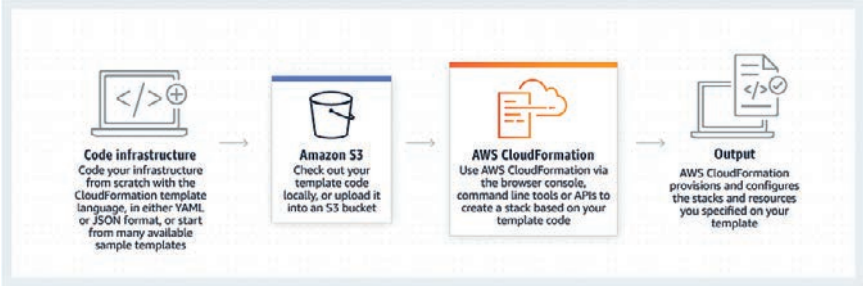


Figure 7.3
AWS CloudFormation (source: https://aws.amazon.com/cloudformation/; © 2024, Amazon Web Services, Inc.)

Other options include the following.

- **Google Cloud Deployment Manager:** Google Cloud Deployment Manager is an infrastructure deployment service that automates the creation and management of GCP resources using templates. These templates, written in YAML or Jinja2, define the desired state of the infrastructure.
  Key features include the following:
  - Integration with other GCP services such as Compute Engine, Cloud Storage, and BigQuery

- Support for declarative configuration using templates
- Version control and reuse of templates

- **ARM templates (Microsoft Azure):** ARM is the infrastructure management service for Microsoft Azure that enables users to provision and manage Azure resources through declarative templates. These templates are JSON files that define the resources and their configurations.

  Key features include the following:
  - Integration with various Azure services, such as Virtual Machines, Azure SQL Database, and Azure App Service
  - Role-based access control (RBAC) for fine-grained access management
  - Template functions and expressions for dynamic resource creation
- **Terraform (by HashiCorp):** While not specific to any cloud provider, Terraform is a popular IaC tool that supports provisioning and managing resources across multiple cloud platforms, including AWS, GCP, and Azure. Terraform configurations are written in HashiCorp Configuration Language (HCL) or JSON.

  Key features include the following:
  - Multicloud support for provisioning resources on AWS, GCP, Azure, and other providers
  - Infrastructure state management and versioning
  - Modular and reusable configurations with modules
- **Ansible (by Red Hat, now IBM):** Ansible is an open source automation tool that includes modules for infrastructure provisioning, configuration management, and application deployment. While it's not focused solely on IaC, Ansible can be used to define and manage cloud resources on AWS, GCP, Azure, and other platforms.

  Key features include the following:
  - Agentless architecture for easy deployment and management
  - Support for YAML-based playbooks to define tasks and configurations
  - Integration with cloud provider APIs for resource provisioning

These alternatives provide similar functionality to AWS CloudFormation for infrastructure provisioning and management, with each offering its own set of features and capabilities suited to different use cases and preferences.

Practicing Disaster Recovery Frequently

Disaster recovery is a critical aspect of CRE, ensuring that businesses can quickly recover their data and operations in the event of unexpected disruptions. AWS offers a range of disaster recovery services and tools to help organizations create robust recovery strategies.

One of the key services in this domain is AWS Backup, which simplifies and centralizes the backup of data across various AWS services. AWS Backup allows users to automate the backup of their EBS volumes, RDS databases, DynamoDB tables, and more. It provides a unified console for managing backups and enables the creation of backup policies, making it easier to adhere to recovery point objectives (RPOs) and recovery time objectives (RTOs).

AWS Disaster Recovery Tools encompass a variety of services and features that help organizations build and test disaster recovery plans. For instance, AWS CloudEndure Disaster Recovery provides continuous replication of on-premises workloads to AWS, facilitating seamless failover in case of a disaster. AWS also offers services such as AWS Site Recovery and AWS Elastic Disaster Recovery (a CloudEndure service), which automate the recovery process to help organizations minimize downtime and data loss.

AWS Import/Export allows businesses to transfer large volumes of data into and out of AWS efficiently. While not solely a disaster recovery tool, it plays a vital role in disaster recovery planning by enabling the rapid transfer of critical data to AWS, ensuring that organizations can quickly access their data in case of a disaster.

AWS DataSync is another valuable tool for disaster recovery, particularly for organizations with extensive data transfer needs. DataSync simplifies and accelerates data movement between on-premises storage and AWS, helping organizations maintain an up-to-date copy of their data in the cloud for rapid recovery.

AWS Snowball takes disaster recovery to another level, especially for organizations dealing with massive datasets. Snowball is a physical device that allows businesses to transfer large volumes of data to and from AWS securely. In a disaster recovery scenario, Snowball can be used to expedite the process of restoring critical data to the cloud.

GCP also offers several services and tools for disaster recovery, including Google Cloud Storage, Google Compute Engine, and Google Cloud SQL. Google Cloud Storage provides highly durable and available object storage, allowing users to store backup data securely with built-in redundancy across multiple locations. Google Compute Engine enables users to create VM instances in different regions and zones, facilitating geographic redundancy for critical workloads. Additionally, Google Cloud SQL offers managed database services with automatic backups, point-in-time recovery, and failover capabilities to ensure data integrity and availability during disaster scenarios. Together, these services form a robust disaster recovery solution that enables businesses to protect their data and applications against various failure events.

Microsoft Azure offers a range of services and tools to support disaster recovery scenarios, including Azure Site Recovery, Azure Backup, and Azure Traffic Manager. Azure Site Recovery provides automated replication and failover capabilities for VMs and physical servers, enabling businesses to replicate workloads to Azure and fail over seamlessly in the event of a disaster. Azure Backup offers scalable, secure, and cost-effective backup solutions for protecting data across on-premises and cloud environments, with features such as incremental backups, encryption, and long-term retention. Azure Traffic Manager allows users to distribute incoming traffic across multiple regions and endpoints, providing high availability and load balancing for critical applications. With these services, Microsoft Azure helps organizations implement robust disaster recovery strategies to minimize downtime and ensure business continuity in the face of disruptions.

Overall, this set of disaster recovery services and tools caters to organizations of all sizes and complexities. These services ensure data resilience, minimize downtime, and facilitate rapid recovery in the face of unforeseen disruptions.

Case Study

To illustrate how to proactively check if your applications and infrastructure are resilient and then optimize them if necessary, let's review a hypothetical case of using AWS FIS, described previously in the chapter, for productively testing applications and infrastructure by injecting faults and disruptions into a cloud environment. Consider Pearl of the Nile, a fictional, successful e-commerce company that relies heavily on its online platform to generate revenue. The company's website, mobile application, and backend services run on AWS, serving millions of customers daily. Ensuring the reliability and resilience of its digital infrastructure is critical to maintaining customer trust and revenue.

Pearl of the Nile faces several challenges related to ensuring the resilience of its systems.

- It needs to identify vulnerabilities and weaknesses in its architecture before they lead to costly outages.
- It wants to implement chaos engineering practices to proactively test its infrastructure's resilience.
- It is looking for a tool to simulate real-world incidents to understand how its systems respond to failures gracefully.

To address these challenges, Pearl of the Nile implements a five-step process.

Step 1. Identifying critical scenarios: Pearl of the Nile collaborates with its DevOps and site reliability engineering (SRE) teams to identify critical scenarios that could lead to service disruptions or performance degradation. These scenarios

include unavailability of an AWS AZ; network latency between services; resource exhaustion, such as CPU or memory, on critical instances; and failures in third-party service integrations.

Step 2. Creating fault injection experiments: Using AWS FIS, Pearl of the Nile creates a series of fault injection experiments to simulate these critical scenarios in a controlled manner. For example, it configures an experiment to randomly disrupt network connectivity between two microservices to mimic network issues. Another experiment simulates an AWS AZ failure by shutting down resources in one of the AZs.

Step 3. Executing experiments: Pearl of the Nile schedules these experiments during off-peak hours to minimize customer impact. The company starts with less-critical experiments and gradually increases complexity and severity as it gains confidence in its systems' resilience.

Step 4. Monitoring and learning: During each experiment, Pearl of the Nile closely monitors the behavior of its systems using AWS CloudWatch, AWS X-Ray, and other monitoring tools. For example, the company analyzes how its systems respond to the injected faults, looking for unexpected failures, performance bottlenecks, or areas where the system can be further optimized. The teams also gather data on incident response times and how effectively automated recovery mechanisms kick in.

Step 5. Continuous improvement: Based on the results of each experiment, Pearl of the Nile iteratively improves its infrastructure and application resilience. In addition, the teams refine their incident response procedures, enhance resource allocation strategies, and optimize configurations to ensure graceful degradation under failure conditions.

By using AWS FIS, Pearl of the Nile achieves the following outcomes: increased confidence in the resilience of its systems, proactive identification and mitigation of vulnerabilities and weaknesses, improved incident response and recovery times, enhanced customer trust, and reduced revenue loss due to unplanned outages.

Summary

Integrating tools that help with automatic failovers, automatic rollbacks, automatic deployments, chaos engineering, incident response, configuration management, immutable infrastructure, and disaster recovery into your workflow may require collaboration among product, leadership, and engineering teams. It's crucial to communicate the value of these initiatives to application teams as engineering-driven efforts aimed at enhancing system reliability, rather than mandates imposed by leadership. This approach fosters a culture of shared responsibility for system health and encourages teams to proactively address potential issues. Ultimately, utilizing a combination of tools and fostering a Lean culture of continuous improvement can lead to more robust, efficient, and reliable cloud-based solutions.

Q&A

Q: Describe the difference between rollback or "n–1" deployments, blue-green deployments, and canary deployments.

N–1 deployments, blue-green deployments, and canary deployments are different strategies used in software deployment and release management. N–1 deployments maintain a fallback environment, blue-green deployments offer parallel environments for switching between versions, and canary deployments roll out new releases to a subset of users for testing and monitoring. Which strategy to choose depends on factors such as risk tolerance, downtime constraints, and the need for early issue detection in your software release process.

- An **n–1 deployment** strategy involves deploying a new version of the software to all but one of the available environments.

The one environment that is not updated is typically referred to as the "n–1" environment, representing the previous version of the software. N–1 deployments are often used as a risk mitigation strategy. By leaving one environment running the previous version, organizations have a fallback option in case any critical issues or unexpected problems arise with the new release. This minimizes downtime and potential disruptions. For example, if you have three production environments (A, B, and C), you would update environments B and C with the new version, leaving environment A running the previous version as the n–1 environment.

- **Blue-green deployments** involve maintaining two separate environments. The current production environment is often referred to as the "blue" environment, and the new version, which is deployed and tested in isolation, is often referred to as the "green" environment. Blue-green deployments are used to minimize downtime and risk during software releases. The green environment allows testing and validation of the new release without affecting the blue environment. Once testing is successful, traffic is switched from the blue to the green environment.
- **Canary deployments** involve deploying a new version of the software to a small subset of users or instances first (the "canaries"), before rolling it out to the entire user base or environment. This allows for gradual testing and monitoring of the new release's performance and stability. Canary deployments are used to detect and mitigate issues early in the release process. By exposing a small number of users to the new version, you can monitor metrics and gather feedback to assess its impact. If issues arise, you can limit the impact to a smaller user group. For example, instead of deploying a new version to all users simultaneously, you deploy it to a small percentage of users or instances, monitor performance and user feedback, and gradually increase the exposure if everything looks stable.

Q: Which cloud providers and third-party companies offer CRE tools?

There are multiple tools and services offered by different cloud providers and third-party companies. Often, choosing which tool to use depends on an organization's specific needs, existing infrastructure, and familiarity with a particular cloud platform. Here are some examples.

- **Amazon Web Services:** AWS offers a broad range of services for CRE, including Amazon CloudWatch for monitoring and logging, AWS X-Ray for distributed tracing, and Amazon EC2 Auto Scaling for auto-scaling infrastructure. AWS provides load-balancing services through ELB and disaster recovery with AWS Backup and AWS Elastic Disaster Recovery, helping to ensure fault tolerance and high availability across cloud environments.
- **Google Cloud Platform:** Google offers its suite of tools and services for CRE, including Google Cloud Monitoring, Google Cloud Logging, and Google Cloud Trace for monitoring and diagnostics. GCP also provides load balancing, auto-scaling, and disaster recovery options similar to AWS.
- **Microsoft Azure:** Azure provides services such as Azure Monitor, Azure Application Insights, and Azure Automation for monitoring, diagnostics, and automation. Azure Traffic Manager and Azure Load Balancer offer load-balancing capabilities. Azure Site Recovery and Azure Backup are used for disaster recovery and backup solutions.
- **Third-party solutions:** Many third-party vendors offer tools that support these CRE practices, providing a unified approach to monitoring, automation, and incident response. Examples include Datadog, New Relic, and PagerDuty, which integrate with AWS, GCP, Azure, and other cloud providers.

These services allow teams to monitor, diagnose, and automatically adjust workloads to maintain reliability and performance at scale. Ultimately, the choice of CRE tools depends on an organization's specific requirements, multicloud strategy, and preferences. All cloud providers and third-party tools have their strengths and weaknesses, so organizations should evaluate them based on their unique needs and goals to ensure the reliability and resilience of their cloud-based systems.

Index

B

C

D

E

F

G

H

I

J

K

L

M

N

O

P

Q

R

S

T

U

V

W

X

Y

Z