RAFAY

PLATFORM ENGINEERING
BEST PRACTICES

Strategies for Developer Autonomy

WHITE PAPER

CONTENTS

Executive Summary

In today's fast-paced technology environment, developers and other cloud users need on-demand access to infrastructure resources to build, test, and deploy applications rapidly. However, navigating complex IT processes to request and configure things like containers, clusters, namespaces, landing zones, AI workbenches, and other cloud environments can hinder productivity with delays, frustration, and cognitive overload. These cloud users are, as a result, held back from delivering the rapid progress the business demands.

The National Institute of Standards and Technology (NIST), defines on demand self-service as follows: **"A consumer can unilaterally provision computing capabilities, such as server time and network storage, as needed automatically without requiring human interaction with each service provider."** Overall, developer autonomy through self-service workflows to provision cloud infrastructure makes clouds easier to consume while enabling innovation and responsiveness. It allows you to move fast and innovate when dependable access to clusters, namespaces, landing zones, and other cloud resources is available to anyone who needs it for development or testing. This includes organizations that still want an operations team to process infrastructure requests, but are open to allowing those teams to heavily leverage automation to streamline their processes and get devs access to infrastructure more quickly.

This is different from "shadow IT", which refers to technology solutions and systems that happen outside the traditional IT structures within companies due to the business desire for greater speed and flexibility. Some estimates suggest that 30-50% of IT spend at many corporations falls into the shadow IT category. Automation provides similar speed and flexibility, without resorting to duplicate, rogue spending and toolsets. And, with the growth in easy-to-acquire AI solutions from large cloud providers, shadow IT now increasingly includes various AI and machine learning capabilities.

Business units can easily procure AI solutions without involving their internal IT departments by leveraging cloud platforms. This can accelerate prototyping and innovation around AI within organizations, but also increase risks if not properly managed as these AI systems are being built and deployed outside of IT's governance. Gartner predicts that through 2022, 85% of AI projects will be considered shadow AI, lacking robust oversight and governance from corporate IT teams.

Rafay believes that modern applications can **power a better future and they deserve a mature, battle-tested and easy-to-use platform** to automate the infrastructure processes that underpin them.

This is even more important now: given the AI "arms race" organizations are in and lessons learned from prior attempts to scale IT infrastructure, IT engineers are eager to build solid foundations for automated self-service workflows that will foster experimentation and scale easily as demands grow.

Rafay believes that modern applications can power a better future and they deserve a mature, battle-tested and easy-to-use platform to automate the infrastructure processes that underpin them. Without that, the promises and business value of these cutting-edge applications built on cloud, IoT and 5G technologies are at risk. In pursuit of this, Rafay helps enterprise platform teams create a modern operations practice to support the increasing demands for agility, scalability, security, and performance placed on their modern infrastructure by the business.

Rafay has developed this guide to look at ways platform engineers can use automation to give development teams more autonomy through self-service, thereby increasing their productivity. Our experience working with customers in every major industry has shown us that autonomy for developers and operations is the result of maintaining platform control and operational efficiency.

How Does Self-Service Relate to Autonomy?

The concept of self-service has emerged as cloud computing has enabled on-demand access to infrastructure resources. At its core, self-service refers to platform users across an organization having the autonomy to carry out key tasks without being dependent on IT operations teams.

For developers and engineers specifically, this means being able to easily gain access to the compute (such as Kubernetes clusters), storage, network, Kubernetes clusters and other cloud services essential for their projects without delays caused by human operators following manual processes. Having rapid, self-service access to infrastructure empowers them to focus their time on coding and creating business value rather than manual configuration or ticketing processes.

But the self-service experience exists on a spectrum - from streamlined ticketing workflows with responsive assistance driving rapid turnaround times, to fully automated end-to-end provisioning without operations staff involvement. The technical mechanisms differ across the spectrum, but the outcome remains aligned - frictionless access for users to cloud environments so innovation moves at the pace of business. This democratic access paired with appropriate oversight represents the spirit of providing developer autonomy through modern IT self-service.

In other words, for some organizations, “self-service” could be a fully automated portal for developers to request resources. For others, it could still be ticket-based with an operations team in the loop, but so responsive due to assistive automation, that the development experience is granted autonomy nearly equivalent to a fully automated experience.

How to Provide Autonomy to Your Cloud Infrastructure Users

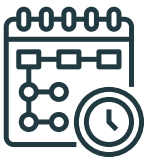
The Challenge

Automated self-service workflows give teams within organizations the autonomy to move at the speed of modern business. But modern clouds are more complicated, leading to bespoke solutions that waste time, cost, and resources, and introduce risk. Platform teams can't always reliably deliver cloud infrastructure as a service to the developers, data scientists, researchers, and other users who depend on it, slowing innovation.

Current cloud platforms provide provisioning and lifecycle management, but lack the tools to provide the needed level of autonomy across teams, applications, or clouds at scale. It is delivering things like clusters, namespaces, or cloud environments as a service, that makes that autonomy possible in the current software development world.

Common Pain Points

Cloud users are often held back by delayed projects, as they navigate complex processes to use cloud resources. Here are some of the common challenges they encounter:



Delays

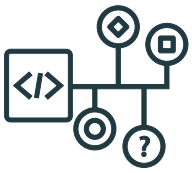
Lack of rapid self-service processes causes delays when users must wait days or weeks to get needed cloud resources. This cripples agility and prompts undesirable shadow IT practices. Some examples of this include:

- Long wait times for infrastructure provisioning stifle innovation and responsiveness. Developers lose momentum when they cannot get environments spun up quickly.
- Developers seek faster alternatives outside official IT channels, building up shadow IT technical debt and risk.
- Delays cause lost opportunities and revenue when new apps/services are slow to market due to infrastructure bottlenecks.

Distraction

Forcing developers and data scientists to constantly learn and manage infrastructure operations distracts users from their core development and testing activities, draining productivity. This results in:

- Time spent figuring out how to configure infrastructure takes away from developing applications.
- Lack of self-service automation forces users to manually interact with low-level cloud APIs and interfaces.
- Cognitive load of managing infrastructure delays projects and diminishes users' focus on driving business value.



Inconsistency

Absence of templatized configurations leads to issues with environment sprawl, version control, compliance, and security when users build resources manually. This manifests itself in several ways:

- One-off manual builds create "snowflake" servers, lacking consistency, compliance, and security best practices.
- Lack of templates or version control causes configuration drift across environments as they are modified independently over time.
- Custom one-off configurations are difficult to manage, replicate, rollback or rebuild if issues arise. Standardized templates are essential.

Waste

Decentralized control results in redundant, orphaned environments and excess resource usage as teams provision dedicated assets without sharing or oversight. The consequences of this are:

- With no central oversight or governance, teams build duplicated environments they think are "theirs", wasting resources.
- These orphaned environments are often forgotten and remain running, accumulating costs.
- Shared infrastructure options can drive up resource usage.

Bottlenecks

Slow processes to move projects into production delay application delivery and time-to-market. The result of which becomes:

- Manual infrastructure provisioning creates bottlenecks slowing down deployments and slow speed to market cedes competitive advantage to nimbler companies.
- Change approval processes and testing environments delay production pushes.
- Business stakeholders wait longer for features and fixes that drive revenue and productivity, such as in the use of GenAI.

The Way Forward

Autonomy requires Control and Efficiency

So, given those challenges and pain points, how do we overcome them to provide the autonomy that will address the broad and complex needs development teams have? It really all comes down to this: control plus efficiency equals autonomy.

Control Your Sprawling Cloud Infrastructure

As organizations rapidly adopt cloud, infrastructure complexity and sprawl often grow unchecked. What started as a few test workloads often proliferates into an unmanageable mess spanning multiple clouds. Teams independently build fragmented, inconsistent resources lacking governance.

However, with the right mix of flexibility and guardrails, infrastructure can scale elastically to meet business needs without compromising reliability, security or efficiency. Proper controls and governance not only enable digital transformation initiatives but also restore IT's capacity to partner on strategy and innovation with the very users they aim to serve. This is the critical lynchpin to delivering on-demand self-service access to cloud infrastructure by developers.

By enabling self-service capabilities for users while layering on enterprise-level oversight, cloud infrastructure can continue growing rapidly but in a disciplined way. Automation frees up IT staff to focus less on manual configuration and more on innovating and optimizing for the business.

Eventually, the architecture combines flexibility and control in a governed hybrid landscape with high-velocity deployment and experimentation built on core platforms, master data, and non-negotiable constraints. This is how cloud complexity is sustainably managed.

CONTROL BEST PRACTICES

As discussed in the previous section, there is no shortage of areas to look at when planning how to provide and maintain the proper amount of control in the development environment. We look at the adoption of reasonable controls being grouped into five key areas of best practices:

1 STANDARDS

Standardizing configurations using infrastructure-as-code techniques brings consistency and prevents divergence across clouds. Common guardrails are applied preventing teams from building unsafe environments. Having pre-approved infrastructure blueprints speeds deployment through golden paths, while ensuring quality.

- Standardized Kubernetes configurations, environment templates, and add-ons across multiple clusters and clouds
- Bi-directional synching and drift detection to reconcile deployed environments with infrastructure-as-code sources of truth like Terraform
- Standardized configs to simplify management and troubleshooting

2 COST TRACKING

Making teams accountable for their cloud spending changes behaviors. Chargeback models tying usage to department budgets incentivize finding efficiencies and eliminating waste. Visibility tools give transparency over consumption and costs.

- Tools to optimize costs and track chargebacks across teams
- Dashboards to track consumption and spending
- Showbacks or chargeback reports to incentivize efficient use of shared resources

3 INTERNAL PORTAL

While simple in concept, the adoption of a centralized portal can have significant positive impacts. The portal becomes the "hub" where all parties involved can find and access the most updated information, tools, templates, and standards.

- An internal portal (for example, via Backstage) for developers and cloud users
- Self-service access to provision infrastructure on-demand increases productivity
- Automation that frees developers from manual processes and excessive cognitive load

4 ADMIN TOOLS

Centralizing access controls and policies ensures adherence to corporate standards. Role-based access manages permissions while robust monitoring provides visibility into the security posture and resource consumption of business units.

- Tools to administer and monitor container environments, AI/ML testbeds, and landing zones
- Add-on management for software required for application, Kubernetes and cloud environment operations
- Simplified deployment and management of complex modern workloads with tailored solutions and access controls

5 POLICY COMPLIANCE

As cloud environments scale rapidly, lack of control and governance can lead to inefficient sprawl and rising risks. Getting back on track requires implementing organization-wide controls covering architecture, security, compliance and costs.

- Enforcement tools for policy compliance with history tracking, audit logs, and cost controls
- Centrally enforced security, compliance, and cost guardrails on all environments and users
- Audit trails for troubleshooting

BENEFITS OF GOOD CONTROLS

Following those best practices can provide thoughtful and measured controls that can yield significant benefits. Better governance of permitted configs from deployment onwards, as well as standardized configurations prevent drift and inconsistency. It's important to ensure changes are controlled through code pipelines. There is less security risk with centralized, role-based access control and auditing, and granular permissions on infrastructure with full auditing decreases the risk profile. Shared accountability of cloud costs with visibility & chargeback, and visibility into consumption and spending coupled with accountable chargeback models optimize efficiency across the organization.

Boost Efficiency of Cloud Operations

As organizations grow their cloud footprint, managing operational complexity becomes a challenge. Environments sprawl across clouds with inconsistent configurations and access controls. This scaling complexity creates inefficiencies like elongated troubleshooting and inflated cloud bills.

Reining in chaos starts with policy guardrails and automation standardization. The question becomes, how do we boost efficiency through managing operational workflow automations (like cluster upgrades), greater resource utilization through use of namespaces, and other methods?

With the right mix of flexibility and constraint, infrastructure can scale elastically to meet business needs without compromising reliability, security or efficiency. Automation and governance not only enable digital transformation but also restore IT's capacity to partner on strategy and innovation. The future is thriving hybrid landscapes with steady oversight.

EFFICIENCY BEST PRACTICES

Attainment of an optimally efficient cloud infrastructure that can grow and scale with your business can be achieved by following best practices:

1 NAMESPACE MANAGEMENT

Multi-tenant clusters are an extremely useful tool for decreasing cloud costs and increasing cluster utilization, if the namespaces within can be properly secured and isolated from one another.

- Tools to manage namespaces across multi-tenant clusters securely and easily.
- Simple processes to grant teams isolated access with controls to prevent interference between workloads.

2 ACCESS CONTROLS

Centralized access controls and policies ensure adherence to corporate standards. Role based access manages permissions while robust monitoring provides visibility into resource consumption.

- Controls for zero-trust kubectl multi-cluster access to users with RBAC.
- Enforced least privilege access with role-based controls and auditing across all clusters.

3 AUTOMATED PROCESSES

Automation and governance not only enable digital transformation but also restore IT's capacity to focus on innovation.

- Automated processes for upgrades and fleet management workflows.
- Reduced operational overhead of cluster maintenance by scripting update processes.

4 CHANGE MANAGEMENT

Pre-approved infrastructure blueprints speed deployment while ensuring quality. Centrally managed configuration updates to ensure ongoing stability and compliance.

- Tools to limit changes allowed by users when deploying.
- Configuration drift prevention by restricting modifications to permitted options only.
- Configuration changes pushed to the entire fleet automatically and safely.

5 TIME TO LIVE CONTROLS

Automated workflows for managing resource use, as well as reclaiming unused resources promptly, help eliminate waste and overspending.

- Time to live controls to eliminate resource waste in 'zombie' environments.
- Automatic recovery of forgotten ephemeral resources based on project status.

BENEFITS OF IMPROVED EFFICIENCY

Tangible and measurable benefits can be obtained with the best practices noted above. Increased efficiency will include streamlined operations due to automation of common cloud management procedures, and an IT staff that can focus less on manual tasks and more on innovation and improvements. This can also result in a Lower Mean Time to Repair, since fewer unique configurations leads to faster troubleshooting. Consistent configurations simplify diagnosing and resolving operational issues. The future is efficient and optimized hybrid cloud management that allows scalability at every opportunity.

Case Studies - Delivering Autonomy through Control and Efficiency

MoneyGram

Driving Autonomy by Streamlining Amazon EKS Operations

After evaluating several vendors, MoneyGram chose the Rafay Cloud Automation Platform to streamline development operations for clusters and namespaces as a service and deliver autonomy to developers. With its deep integration with Amazon EKS, Rafay delivered a single pane of glass for global controls, visibility and monitoring of all Amazon EKS clusters and automated cluster lifecycle management of 40+ workloads, including cluster provisioning, scaling, and one click Amazon EKS upgrades fleet-wide.

Rafay also allowed MoneyGram to create 100+ cluster blueprints to efficiently define, procure, and enforce standard cluster configurations as a service across development and production environments. Further, Rafay natively connected to MoneyGram's Okta account for instant single sign-on for all developers, operations, and support personnel and enabled role based access control (RBAC) and isolation boundaries to be easily defined and enforced.

Guardant

Empowering Autonomy for Oncology Research

Guardant Health manages HPC clusters with Kubernetes in order to provide time-sensitive results to tests that support oncology research. Since deploying Rafay, there have been no issues with reliability, and making applications are portable as needed. Rafay's cluster blueprints help internal teams standardize clusters across the company and deploy applications faster. Additionally, Rafay enables Guardant Health's developers to have the autonomy to quickly build, test and deploy applications, supporting the development team's robust QA and production pipelines. Increased control through templates, as well as the ease of controlling access and creating isolation between teams were keys to their success.

The organization's development operations practice now runs more efficiently, only requiring a fraction of the time and resources to operate and maintain, allowing the Guardant Health team to focus on their innovation and not on their Kubernetes infrastructure.

Fortune 50 Consulting Company

From Rancher to Rafay: Orchestrating Autonomy in Cloud Operations

Rapid customer growth brought new challenges for the cloud operations (ops) team. It became clear that the ops team needed to re-evaluate their Kubernetes technology stack due to scaling challenges with Rancher, their existing solution. The number of Rancher instances needed to manage their current solution kept increasing because each was able to control only a small segment of clusters. The growing installation, configuration, and ongoing maintenance of these dedicated servers led to project delays and complexity when managing isolation boundaries.

With Rafay, the firm expanded services to Amazon quickly and took advantage of comprehensive visibility and management across their fleet of heterogeneous clusters to provide autonomy for hundreds of customers. The ops team automated provisioning and lifecycle management for both clusters and applications which increased efficiency by reducing lead time to build infrastructure and deploy applications. Furthermore, Cluster Blueprints helped control consistency and enforced standards for security policies and software add-ons via Git. This eliminated snowflake clusters which, in turn, reduced the MTTR and the cost of support of said clusters.

Conclusion: Autonomy = Control + Efficiency

As cloud environments grow rapidly in complexity, lack of governance leads to inefficient sprawl and rising risks. Implementing organization-wide controls brings order - from architecture standards to security policies to cost accountability. Standardizing configurations using infrastructure-as-code techniques brings consistency and prevents divergence across clouds. Common guardrails are applied preventing teams from building unsafe environments.

Proper controls can help transform the cloud infrastructure from “Wild West” to well-run core business capability even as complexity continues growing. With the proper guardrails in place, organizations can build as much cloud as they need. In fact, Rafay customers have have achieved as much as **5x cloud growth** without growing the associated overhead, supporting a flourishing business.

5X
Cloud Growth

76%
MTTR Reduced

4X
Deployments Increaced

Policy enforcement and governance guardrails are critical. Self-service capabilities must be paired with access controls, quotas, approvals etc. to prevent sprawl, ensure regulatory compliance, and reduce the impact of outages. Automation and orchestration tools codify and automate provisioning and management of infrastructure. Rafay customers have **reduced MTTR by 76%**, improving efficiency and availability.

By implementing self-service automation with governance, you can unlock user productivity, increase collaboration, and deliver applications faster without compromising policy, security, or your budget. Rafay users operate with more autonomy, moving faster and **increasing deployments by 4x** while staying aligned to business priorities.

Examining real-world case studies gives a peek into how companies are taking advantage of Rafay’s solutions to maximize their investments of all manner of resources in order to serve their own customers and realize profitability and growth. These examples go beyond abstract concepts and principles and show how Rafay customers are achieving success.

This guide examined best practices for empowering developers, data scientists, testers, and engineers with self-service access to cloud resources. By focusing on the idea that autonomy is the result of investments in control and efficiency, we broke down the common pain points encountered by cloud platform users and engineers, what the future holds for cloud infrastructure technologies, and looked at some practical solutions in the form of best practices to achieve a well-managed cloud environment.

Learn More About Rafay Systems

RAFAY

Rafay Systems, Inc. | 530 Lakeside Dr. Ste 220 Sunnyvale, CA 94085 | 669.336.4800 | rafay.co