

Block's Blueprint: solving the infrastructure as code learning curve

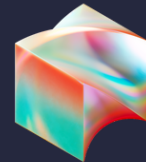
HIGHLIGHTS



92% reduction
in Terraform PR review rate



80% less time
spent on manual reviews



BLOCK

About Block

Block is a global technology company with a focus on financial services. Made up of Square, Cash App, TIDAL, and TBD, they employ thousands of software engineers and ship over 500 Terraform PRs daily.

Challenge

Block's developer teams faced a complex infrastructure configuration experience that slowed developer velocity and caused misconfigurations, which could lead to service interruptions and excess manual reviews.

Outcome

By implementing Resourcely, Block's Cloud Foundations improved developer efficiency and experience, reduced manual reviews, and eliminated accidental infrastructure deletions.

Existing Challenges

After migrating more than 1,000 applications from on-premise to the cloud in recent years, Block's Cloud Foundations Team faced a new set of challenges around managing cloud infrastructure. Early adopters were proficient in cloud services and infrastructure-as-code, but as they rolled out access to hundreds of developers three problems became apparent:

- The learning curve for Terraform was too steep
- Developer experience and velocity were suffering
- Infrastructure was inconsistent and lacking in best practices

Developer velocity is of paramount importance to Block, and they knew providing a self-service experience would reduce the noise and help developers ship faster.

While they did provide some opinionated Terraform modules in conjunction with their Cloud Security and Cloud Economics teams, developers still needed a good understanding of Terraform to use them. As the number of components in a Terraform module increased, so did the complexity of maintaining them (the $n!$ problem).



We wanted to create an environment where developers don't have to worry about the complexities of infrastructure. With Resourcecly, they can simply select the resources they need, fill out a form, and hit submit—no need to dig into Terraform code or become experts in cloud platforms.

Bill Townsley, Director of Cloud Foundations at Block



Even though the Foundations team provided opinionated templates, they still had to review hundreds of pull requests each month. For a platform team of 9, this was unsustainable.

Apart from giving developers an easier and faster way to configure infrastructure, Block was also dealing with a misconfiguration problem. Developers would unknowingly deviate from approved configurations and ship bad Terraform. One misconfiguration was the accidental deletion of resources, an easily avoidable but common problem encountered at many companies. If a customer-facing application was impacted, it could stop Block from taking payments or prevent a merchant from selling something.

The Foundations team mitigated this risk by mandating Terraform PR reviews that touched critical applications. This was a necessary but inefficient practice because not all deletions of infrastructure were harmful. The team took rotations, and every day one team member was dedicated to manual PR reviews.



Before Resourcely, our team had to manually review dozens of pull requests every day, many of which were low-risk but still required our attention. It was a time-consuming process that often took up an entire day, leaving us little room to focus on more critical tasks. The manual reviews created a bottleneck, with other teams having to wait for our approvals before they could move forward.

Kent Broadbent, Senior Software Engineer at Block



The Cloud Foundations team calculated that they were wasting the time of 1 full-time employee stuck in manual review rotation, and slowing down developers by requiring them to use Terraform and then having their PRs wait in a manual review queue. When misconfiguration did happen and customer-facing infrastructure was impacted, the cost could be in the hundreds of thousands of dollars.

They set out to solve the root of their three problems: abstracting away Terraform code, preventing harmful misconfiguration from making it into production, and giving developers the tools to deploy infrastructure faster and with confidence.

Why Resourcely

The Block team discovered Resourcely, a platform for streamlining infrastructure as code configuration and governing cloud resource changes, and immediately knew it could make an impact. With Resourcely's Blueprints and Guardrails, they saw a tool that could help them solve their problems, while adopting at a pace that made sense to them.

Blueprints

Block looked first to Resourcely Blueprints to solve their highest impact problem: developer velocity. At their scale, Block understood that they were wasting valuable engineering resources on a poor infrastructure deployment experience. The most common problems impacting developer experience were:

- **Discoverability** - engineers were having a hard time finding the services they should use, with the customized options they wanted
- **Abstracting Terraform** - developers didn't want to be Terraform experts, and it was slowing them down

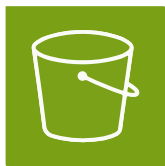
The Foundations team knew that giving developers a self-serve experience would also result in standardized infrastructure and less noise when it came time to review manual requests.

Our PR review rates have improved by 12x with Resourcecy Blueprints, unlocking developers to ship faster across the organization.

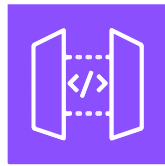
Kent Broadbent, Senior Software Engineer



Resourcecy Blueprints are organized in the Shopping Cart, making it easy to categorize, tag, and discover the service combinations that engineers were looking for. Starting small, Block created some basic Blueprints and proved that developers could find them faster via search. From there, Block realized the full power of Resourcecy Blueprints by fully customizing Blueprints for common AWS services:



Amazon S3



Amazon API Gateway



Amazon Aurora



Amazon RDS



Amazon DynamoDB



Amazon ElasticCache



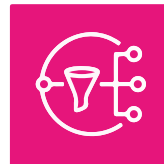
Amazon KMS



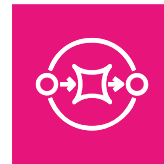
Amazon Lambda



Amazon Secrets Manager



Amazon SNS



Amazon SQS

...as well as other tooling such as PagerDuty and Datadog.

Block views Blueprints as a paved road to production that reduce the surface area that engineers need to touch when deploying infrastructure. They have now rolled out Resourcecy to all of their 3,800 Terraform repos, meaning that they can be used for all of Block's applications.

Guardrails

After proving out their use case for Blueprints, Block looked to Resourcecly Guardrails as the answer to a problem particularly impacting the platform team: potentially harmful misconfiguration making it into production. Their software teams were shipping infrastructure at such a high volume that the Cloud Foundations team of 9 was having trouble reviewing all of the team's PRs. As part of their security team's tooling they had Terraform linting rules in place, but they didn't cover the infrastructure use cases Block was looking for.

The team rolled out Guardrails in a 3-stage process. They started with 19 repos across 3 teams, where they implemented several Guardrails. Here is an example a Guardrail they are using:

```
GUARDRAIL "[Misc] Disallow Deletions"  
WHEN CONTEXT environment MATCHES ANY REGEX ["^prod"] AND DELETING * AND  
NOT pagerduty_* AND NOT datadog_* AND NOT null_resource  
REQUIRE APPROVAL @CloudFoundations
```

This Guardrail prevent deletion of resources in production environments, unless those deletions are for inconsequential tools such as PagerDuty or Datadog. After a couple weeks of testing with a significant reduction in manual PRs because of the Guardrail, the team rolled Resourcecly out to the rest of their critical applications (stage 2) and eventually to all of their Terraform repos (current state).

Resourcecly gave us an easy way to protect our cloud from unwanted changes such as accidental deletion, improper IAM, or public access to sensitive data. They integrate nicely with Blueprints so that our developers get feedback while they are configuring resources, instead of waiting hours or days for feedback from our team.

Kent Broadbent, Senior Software Engineer



Block estimates that Guardrails have reduced the Cloud Foundations team's manual review time by 80% since adopting Resourcecly while dramatically cutting accidental outages from misconfigured IaC.

Looking forward

With Resourcecly, Block's development teams are no longer burdened by slow cycle times that Terraform necessitated. PR manual review rates have improved by 12x, with developers reporting much higher satisfaction rates when using Resourcecly Blueprints.

Resourcecly has given the Cloud Foundations team at Block the ability to control their cloud infrastructure, preventing bad configuration from ever making it to production while unburdening them from hours of manual reviews every day.

Resourcecly has unlocked my team to work on higher value tasks, while giving us the confidence that our infrastructure is stable and secure. Our developers are empowered to ship faster, and they are much happier with the experience of deploying with Resourcecly.

Bill Townsley, Director of Cloud Foundations



Meet your Resourcecly Superheroes!



Bill Townsley, Director of Cloud Foundations

Bill started his career as a SCUBA diver and programmer in support of oceanographic research, then transitioned to Systems and Network Administration during the "dot-com boom." Frustrated with the inefficiencies of managing physical hardware, he fully embraced the Cloud in 2015. Bill joined Block in June of 2022 and loves the company's people, technology, and mission.



Kent Broadbent, Cloud Foundations Engineer

Kent has been working as a software engineer since 2009. He has been part of several large-scale cloud migrations and is an AWS Certified Solutions Architect - Professional and GCP Associate Cloud Engineer. Kent has worked at Block since 2021 and enjoys interesting problems he gets to work on as the company expands its presence in the cloud.