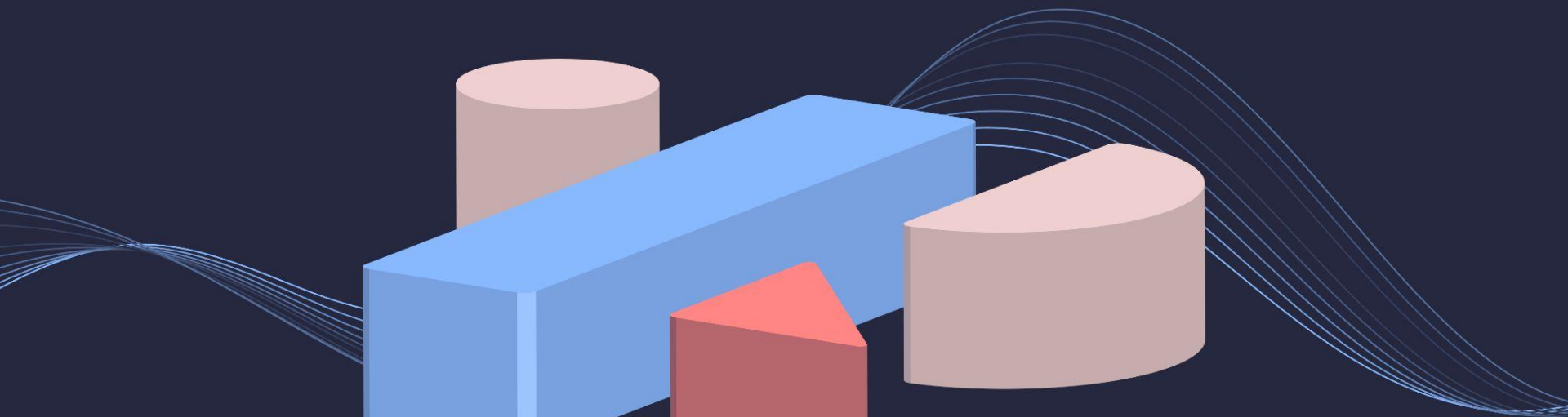




Scale cloud configuration



Scaling cloud infrastructure is hard...

- infrastructure as code learning curve
- cloud services complexity

Results in millions in impact...

- slow, frustrated developers
- excess cost (human capital and cloud spend)
- misconfiguration in production → lack of governance

No vendor is solving it today

Configuration is largely DIY...

- Homegrown tooling
- Terraform-based with a long tail of cloud services
- Very little configuration automation

And not developer-friendly...

- Security tossing vulnerabilities over the fence
- Platform teams stuck in ops

Configuration platform for scaling cloud infrastructure



Resourcemeta Configuration Platform

Prevent problems

Blueprints = Paved paths to production

Guardrails = Policies and rules

Fix problems

Campaigns = Facilitated remediation

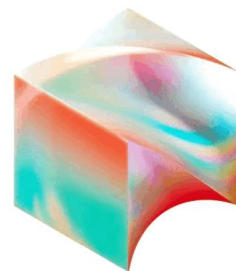


Resourcely has unlocked my team to work on **higher value tasks**, while **giving us the confidence that our infrastructure is stable and secure**.

Our developers are **empowered to ship faster**, and they are much happier with the experience of deploying.



Bill Townsley, Director of Cloud Foundations



BLOCK

Prevent

Blueprints

Customizable paved roads for developers to configure cloud infrastructure

- Improve developer experience and velocity
- Simplify complexity and control how infrastructure is configured
- Scale infrastructure as code adoption

Configure Blueprint Content

💡 Learn more about blueprint authoring by reading our [docs](#).

```
2 constants:
3   | __name: "{{ name }}_{{ __guid }}"
4   | ---
5
6 resource "okta_group" "{{ __name }}" {
7   name = "{{ name | desc: \"New group name\" }}"
8   description = "{{ description | desc: \"New group's purpose\" }}"
9 }
10
```

Preview Developer experience Reset Form Data

Error! 💡 This is a preview of the fields that your developers will see and interact with as they create new resources.

Okta group

General

Name *

New group name

Description *

New group's purpose

Enforced by guardrail [Okta naming convention](#) applied to this blueprint
This option requires approval from default and will be pending their review.

Proceed 🔑

Prevent

Guardrails

```

[Resource Instance] my-pull-request #153
Changes from all commits - File filter - Conversations - Jump to -
csreuter committed 2 weeks ago

45 main.tf
5   foo = var.project
6   }
7 }
8 +
9 + resource "google_compute_instance" "my-gpu-instance_aSrPGiVc9MazQ5FZ" {
10 +   name      = "my-gpu-instance"
11 +   machine_type = "a2-highgpu-4g"
12 +
13 +   tags      = ["my-gpu"]
14 +   zone     = "us-east1-b"
15 +   depends_on = [google_compute_disk.my-gpu-disk]
16 +   boot_disk {
17 +     initialize_params {
18 +       image = "cos-105-105-105-105"
  
```

resourcelybot / Resourcely Guardrails
Resourcely Guardrail

Allowed compute instance types
Requires: type IN ["e2-*,*"]
Category: Cost Efficiency
Contact: security

```

> repo root (dev)
> repo root (prod)
  
```

ADMINISTRATOR

Guardrails

Create and manage guardrails that govern how cloud resources are created and changed

Search by
Name, description

Filter	CATEGORY	NAME	PROVIDER	LAST UPDATED ON
PROVIDER	Cost Efficiency	Allowed compute instance types	Google	2 days ago
<input type="checkbox"/> Amazon	Best Practices	S3 bucket policy not ACL	Amazon	2 days ago
<input type="checkbox"/> Google	Best Practices	QOP Storage Policy not ACL	Google	4 weeks ago
<input type="checkbox"/> Github				
CATEGORY	Best Practices	Storage naming convention	Google	2 days ago
<input type="checkbox"/> Access Control	Access Control	Approval for making storage public	Google	2 days ago
<input type="checkbox"/> Best Practices	Access Control	Approval for making S3 public	Amazon	2 days ago
<input type="checkbox"/> Cost Efficiency	Best Practices	Allowed compute image types	Google	2 days ago
	Best Practices	S3 naming convention	Amazon	2 days ago

Light mode

Policies built for cloud infrastructure

- Make sure bad configuration can't be merged to production
- Prevent incidents, outages, and breaches
- Control cloud costs
- Maintain platform stability and security

Fix

Campaigns

Easily remediate issues in production

- Define your infrastructure standards or best practices
- Identify existing resources that violate your rules
- Manage and implement fixes to your misconfigured infrastructure

Identifying campaign targets

Scanning repositories:

- Resourcecly-inc/terraform/repo1
- Resourcecly-inc/terraform/repo2

Searching for resources in the above repositories that violate the rules of **4 selected Guardrails**.

⚡ Found **125 violations** of selected guardrail rules.

[< Edit scope](#)
[Manage findings](#) >

[Download CSV](#)
[Manage campaign](#) ▼

- || Pause campaign
- ☑ Remediate all
- ✕ Deactivate

STATUS	RESOLVED FINDINGS	GUARDRAIL NAME	CATEGORY
In violation ▼	⚡ 48 of 50	No admin IAM	Access Cor
In violation ▼	⚡ 20 of 46	Include tagging	Best Practices Amazon resource_name
In violation ▼	⚡ 14 of 28	Require multi AZ	Globalization Azure resource_name
Remediated ▼	✔ 1 of 1	Add S3 AES256	Encryption Amazon resource_name

Rows per page: 10 ▼ 1-4 of 4 < >

Where Resourcely fits



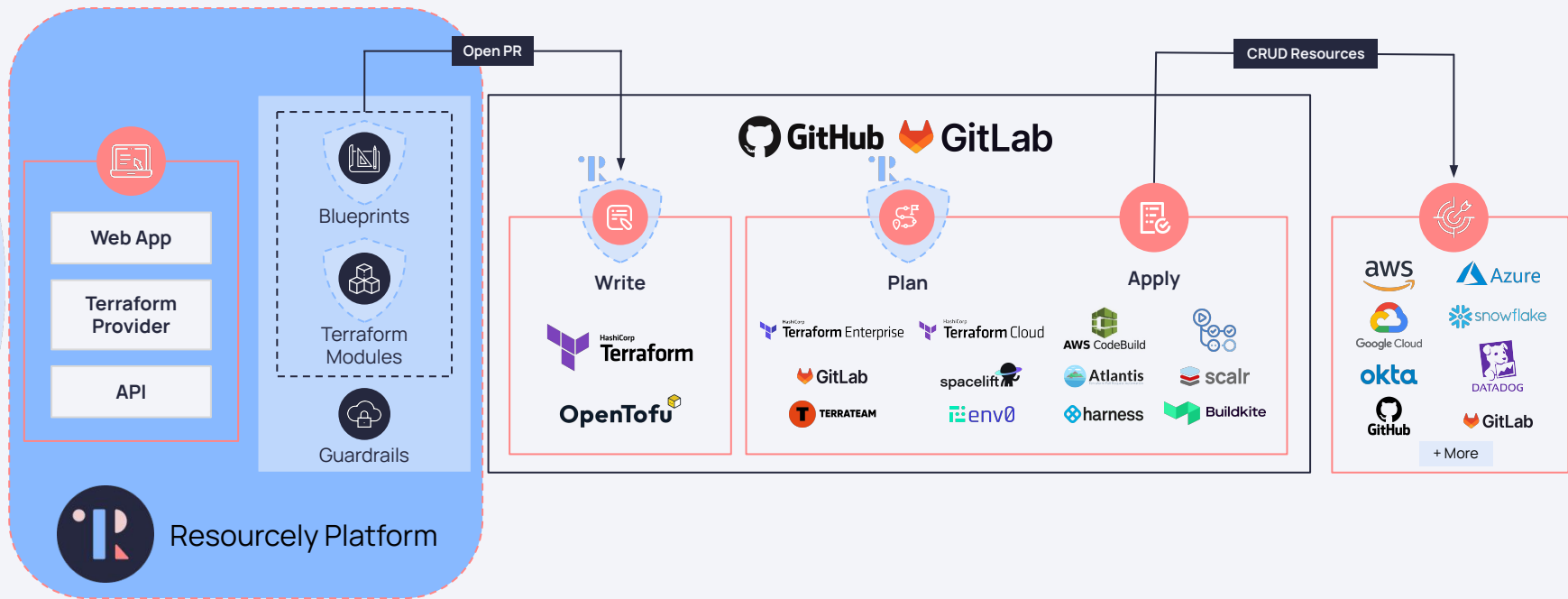
Security



Developers



DevOps



Scaling cloud infrastructure with Resourcely

Block has thousands of developers, shipping Terraform changes to the tune of 600 PRs/week. The problem? The Terraform learning curve was too high.

Pain

- Deploying infrastructure was slow due to Terraform complexity and manual reviews
- Long review queues caused development delays
- Terraform misconfigurations led to customer-facing outages

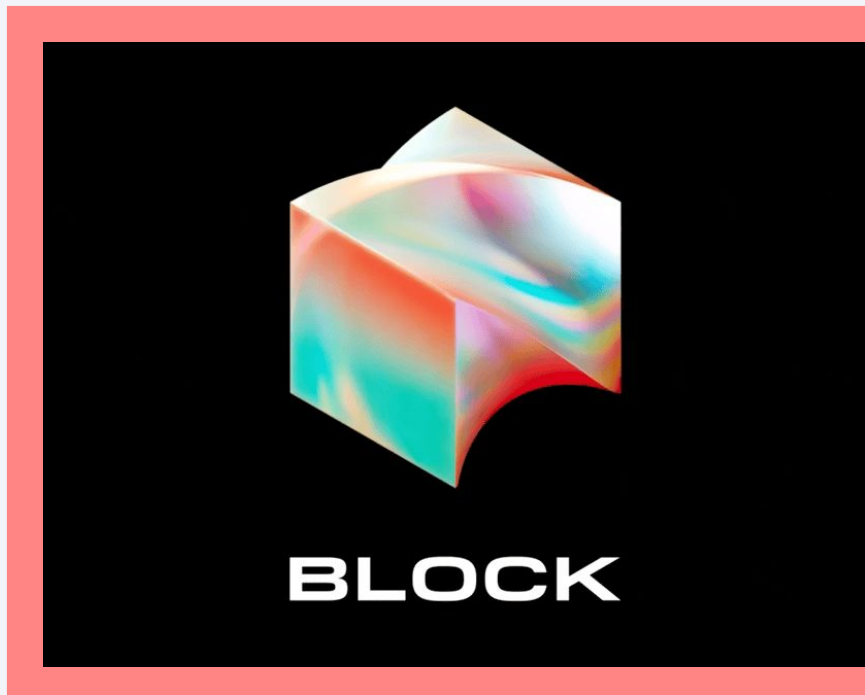
With Resourcely

92%

reduction in PR
review rate

80%

less time spent on manual
PR reviews



Use Cases

Description

Blueprints

Guardrails

FinOps

Reduce costs/control cloud spend

Cost-conscious default and suggestions for compute-intensive infrastructure

Conditional Instance sizing
Control storage type
Multi-AZ restrictions
Enhanced logging/monitoring approvals

Identify & Access Management

Ensure proper permissioning

Make IAM role and policy generation easier and faster

Limit policy permissions
Require approval for new or permissive roles

Kubernetes Configuration Automation

Allow users to deploy containers

Give developers the ability to leverage Kubernetes without making mistakes

Enforce logging, require approvals

Data encryption

Mitigate breaches

Built-in encryption for blob storage, block storage, or managed databases

Require encryption for any infrastructure as code PRs

Deletion prevention

Stop incidents/stability issues

n/a

Require approval for inappropriate deletion of resources

AI

Allow AI use without risks

Fast start for AWS/GCP/Azure AI and ML services - use AI programmatically without infrastructure expertise

Govern foundational models used, GPU cost and amount, data inputs

Appendix

Additional Resources

Live Demo: <https://www.resourcely.io/schedule-demo>

Demo Library: <https://www.resourcely.io/demo-library>

Docs: <https://docs.resourcely.io/>

Free Trial: <https://portal.resourcely.io/>

Proof of Value: <https://www.resourcely.io/pov>



Resourcely

Founded by



Travis McPeak
CEO



Alaeddin Almubayed
CTO

