CLOUDBOLT

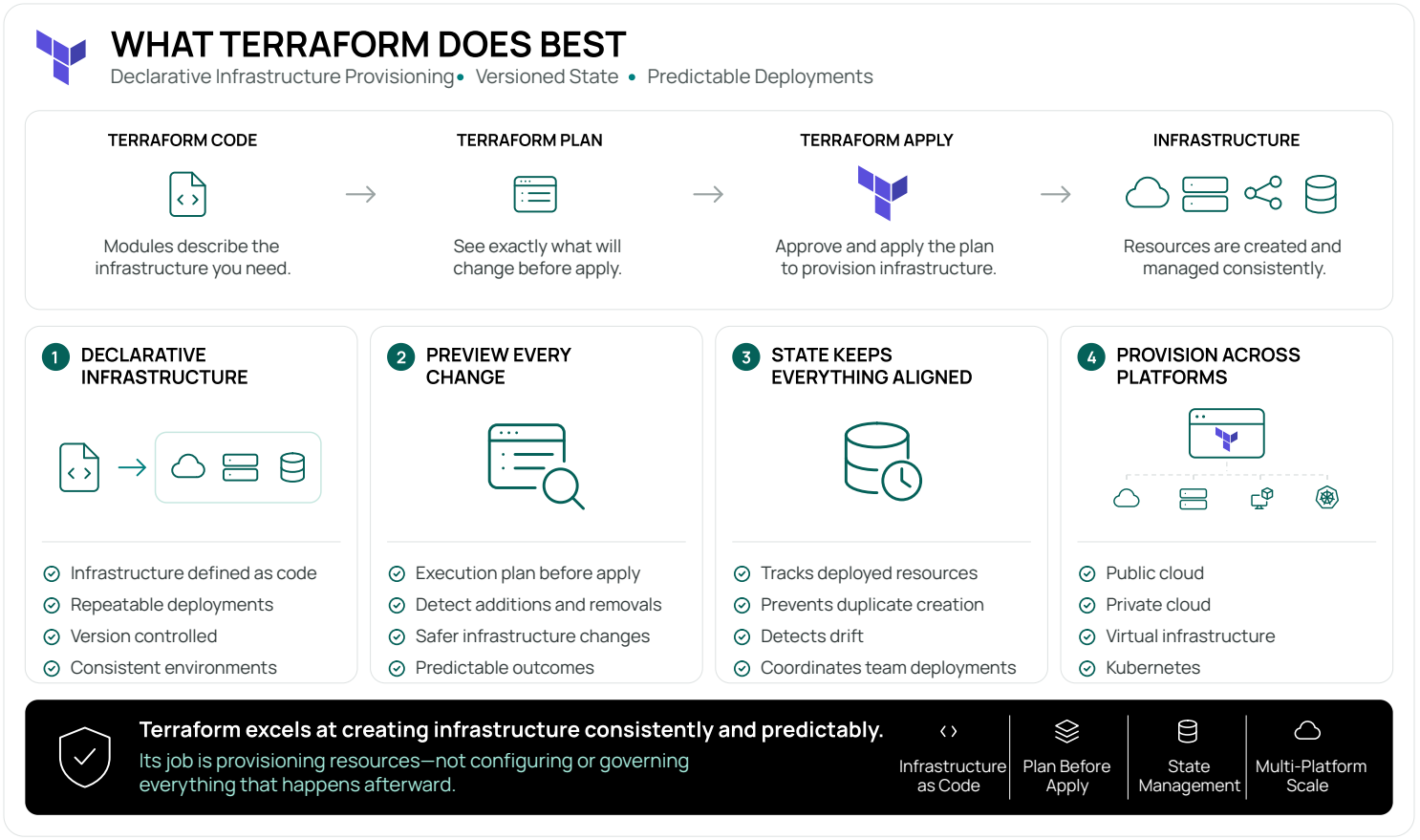
COMPLEMENTARY STACK SERIES: PART 3

Terraform and CMP: Provisioning Is Not the Same Job as Governing

The Terraform version of the pitch is different from the Ansible version, but it lands in the same place. Someone points to HCP Terraform's no-code modules and Sentinel policy checks and asks why a cloud management platform is still needed when Terraform already has a self-service front end and a way to enforce guardrails. The honest answer is that Terraform's newer self-service and policy features are excellent at the job Terraform has always done: turning infrastructure into code and applying it consistently. That job stops at the edge of what Terraform itself provisions, which is honestly a narrower boundary than a full governance layer needs to cover.

WHAT TERRAFORM ACTUALLY DOES WELL

Terraform's core strength is provisioning infrastructure as versioned, declarative code rather than a sequence of manual steps. A module describes the resources a workload needs, a plan shows exactly what will change before anything happens, and state tracks what Terraform believes exists so the next run knows what it is working with. HCP Terraform centralizes that state, coordinates runs across a team, and prevents the state corruption and drift that plagued Terraform when everyone kept their own local copy. For the specific job of standing up and modifying infrastructure predictably, at scale, across clouds, that is a genuinely hard problem solved well.



WHAT NO-CODE MODULES AND SENTINEL ADD…AND WHAT THEY DON'T

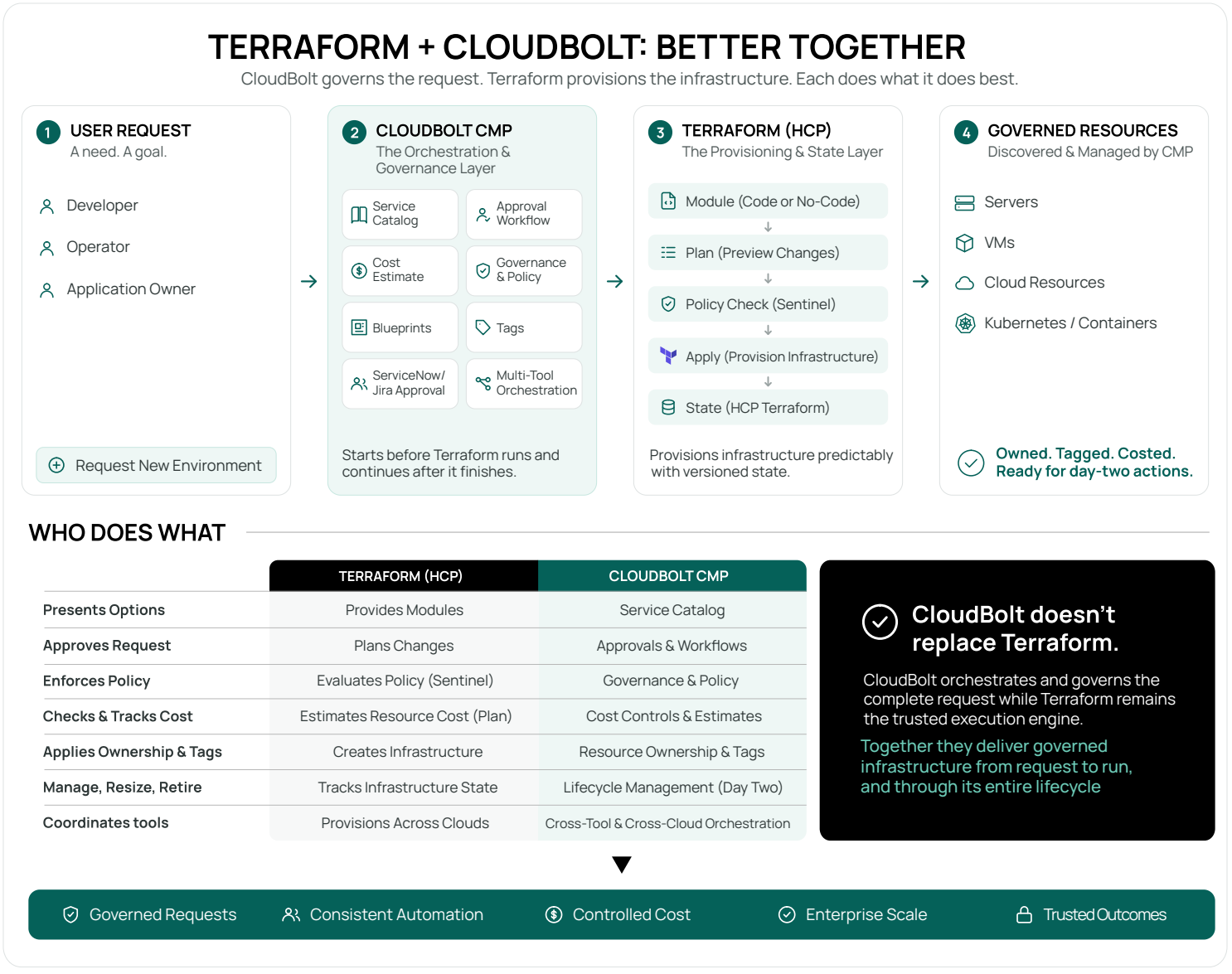
No-code provisioning lets a platform team publish an approved module behind a simple form, so a requester can deploy one specific, pre-vetted thing without writing HashiCorp Configuration Language. It is a real improvement for exactly the audience it targets: people who need one known-good resource, fast. The access model underneath is scoped to Terraform's own permission system, not a general one. Setting up no-code provisioning requires membership on a team with manage-all-projects, manage-all-workspaces, or project-admin permissions, and every workspace it creates still lives inside HCP Terraform's own state and RBAC. Sentinel works the same way: it is a genuinely capable policy-as-code engine, but it runs between a specific plan and apply, evaluating that one Terraform run against that one set of policies. It has no visibility into a request that also touches Ansible, a ticketing system, or a budget approval that lives outside Terraform entirely.

WHERE CMP PICKS UP

A cloud management platform's job starts before Terraform runs and continues after it finishes. CloudBolt lists Terraform among its 200-plus integrations, and blueprints that include Terraform Plan Actions let CMP discover the servers a Terraform run provisions and pull them into a governed resource record automatically. Ownership, tags, and cost tracking get applied to that resource going forward, and day-two actions such as resizing, decommissioning, or extending it are handled through CMP the same way regardless of whether Terraform, a cloud API, or another tool originally created it. Terraform's state file answers what exists. CMP answers who owns it, what it costs, and what happens to it next, questions a Sentinel policy was never built to track once the apply finishes.

THE PRACTICAL PAIRING

A CMP blueprint can call a Terraform no-code module to do the actual provisioning exactly as a platform team already built it, then take over lifecycle management the moment that module finishes running. Terraform still owns the plan, the apply, and the state. CloudBolt still owns the approval chain that got the request there, the cost estimate the requester saw before committing, and the ongoing record of who is responsible for the result. Neither tool has to be rebuilt to make that handoff work, because each one is already doing the job it was designed for.



WHO DOES WHAT

	TERRAFORM (HCP)	CLOUDBOLT CMP
Presents Options	Provides Modules	Service Catalog
Approves Request	Plans Changes	Approvals & Workflows
Enforces Policy	Evaluates Policy (Sentinel)	Governance & Policy
Checks & Tracks Cost	Estimates Resource Cost (Plan)	Cost Controls & Estimates
Applies Ownership & Tags	Creates Infrastructure	Resource Ownership & Tags
Manage, Resize, Retire	Tracks Infrastructure State	Lifecycle Management (Day Two)
Coordinates tools	Provisions Across Clouds	Cross-Tool & Cross-Cloud Orchestration

CloudBolt doesn't replace Terraform.

CloudBolt orchestrates and governs the complete request while Terraform remains the trusted execution engine.

Together they deliver governed infrastructure from request to run, and through its entire lifecycle

Governed Requests · Consistent Automation · Controlled Cost · Enterprise Scale · Trusted Outcomes

THE BOTTOM LINE

HCP Terraform's no-code modules and Sentinel policies make Terraform easier and safer to use, which is exactly what a provisioning tool should keep getting better at.

Neither one turns Terraform into a system that governs requests spanning multiple tools, tracks ownership after the apply, or enforces cost policy outside its own state. That is still a separate job, and it is the one this series looks at from a third angle next: internal developer platforms like Backstage, where the overlap with CMP is more about the catalog than the execution.

Talk to CloudBolt cloudbolt.io/demo