

— COMPLEMENTARY STACK SERIES: PART 5, FINAL

The Complementary Stack: Any Combination, One Constant

Across this series, the same shape kept showing up. Ansible executes automation reliably. Terraform provisions infrastructure as code. Backstage catalogs services and standardizes how developers create them. Not every organization runs all three, and there is no reason to force that. A team built mostly on managed Kubernetes might never touch Ansible. A shop with little in-house application development may never need Backstage at all. What does not change based on that mix is the fourth layer. Whichever combination of execution, provisioning, and portal tools an organization ends up running, something still has to govern the requests that cross between them, track ownership and cost after the fact, and enforce approvals. That is the layer this series keeps landing on regardless of which of the other three are actually in play, which is why a cloud management platform is the constant in this stack, not just one more optional piece alongside the rest.

THE FOUR LAYERS, AND WHY YOU DON'T NEED ALL OF THEM

Configuration management, Ansible's category, executes tasks against systems that already exist. Infrastructure as code, Terraform's category, provisions the resources those systems run on. An internal developer platform, Backstage's category, catalogs what exists and standardizes how new things get created. Most organizations run some combination of these, not all three at once: a team on managed Kubernetes might lean on Terraform and skip Ansible entirely, a small platform team might build golden paths without ever standing up Backstage, and a shop doing heavy on-premises configuration work might never adopt infrastructure as code at all. CloudBolt CMP sits above whichever of these an organization actually has, coordinating the approvals, governance, and cost controls that none of them were built to own. Swap Ansible for Puppet, add Terraform later, retire Backstage for a simpler portal: CMP's job does not change, because it was never tied to which specific tools sit underneath it.

COMPOSABLE PLATFORM ENGINEERING

Use only the layers you need. Govern them all the same way.

EVERY ORGANIZATION IS DIFFERENT

☑ Some use Ansible

☑ Some use Terraform

☑ Some use Backstage

☑ Many use only one or two

OPTIONAL LAYER



Ansible
Execution Layer

Runs configuration and automation reliably against systems that already exist.

OPTIONAL LAYER



Terraform
Provisioning Layer

Provisions infrastructure as versioned, repeatable code.

OPTIONAL LAYER



Backstage
Developer Portal Layer

Catalogs services and standardizes how developers create new ones.

EVERY ORGANIZATION STILL NEEDS

☑ Governance

☑ Approvals

☑ Cost Controls

☑ Ownership

☑ Lifecycle Management

☑ Audit & Compliance



CLOUDBOLT CMP

Governance Layer – The Constant

✓ ALWAYS REQUIRED



Approvals

Policy-driven approval workflows



Cost Management

Estimate, track, and control cost at every step



Ownership

Identify and track owners and accountability



Policy & Compliance

Enforce governance policies and compliance rules



Lifecycle Management

Manage resources from request through retirement



Orchestration

Orchestrate tools and workflows end-to-end



Integrations

Connect to 200+ tools and platforms

Governs requests, enforces policy, controls cost, and tracks ownership across whichever tools are in play.

YOUR ENVIRONMENTS & SYSTEMS



AWS



Kubernetes



Azure



On-Premises



Google Cloud



ServiceNow



VMware



Jira



Nutanix



And More

☑ Governance is the constant. Everything else is a choice.



BEST-OF-BREED WINS

Composable architecture lets you choose the right tools today—and change them tomorrow.

Swap tools without rewriting workflows →

Replace Ansible with Puppet, Chef, or something else →

Replace Terraform with Pulumi, CDK, or another option →

Replace Backstage with another portal or custom solution →

CloudBolt CMP continues governing every request

— WHY BEST-OF-BREED IS WINNING THE ARGUMENT

This is not just CloudBolt's framing. Independent analysis of platform engineering going into 2026 describes the winning pattern as composable: assembling a stack from best-of-breed components connected by thin integration layers, with no single vendor lock-in, so any one piece can be replaced without rewriting the rest of the platform. The same analysis is explicit that the alternative, buying one platform that claims to do everything, has failed for most organizations that tried it. Composable does not mean running every category either. It means running only the layers a given workload actually needs, and keeping one consistent governance layer over whatever that subset turns out to be, which is exactly the role CMP fills.

— THE ONE-SENTENCE VERSION OF EACH TOOL

A defensible budget conversation does not require running every layer, just an accurate sentence for whichever ones happen to be in play, and one sentence that holds regardless of the combination.



ANSIBLE AUTOMATION PLATFORM

Ansible Automation Platform, if it is part of the stack: the execution layer that runs configuration and automation reliably, at scale, against systems that already exist.



TERRAFORM

Terraform, if it is part of the stack: the provisioning layer that turns infrastructure into versioned, repeatable code.



BACKSTAGE

Backstage, if it is part of the stack: the developer portal that catalogs services and standardizes how new ones get created.



CLOUDBOLT CMP — ALWAYS

CloudBolt CMP, always: the governance layer that orchestrates approvals, cost, and ownership across whichever of the above an organization actually runs.

The first three are optional, and which ones show up depends entirely on what an organization has already built. The fourth is not optional in the same way, because whatever combination ends up in place still needs something to govern it.

— HOW TO PUSH BACK ON A PREDATORY PITCH

Every article in this series ran the same diagnostic against a vendor's own documentation, and it holds up as a general test no matter which tools an organization runs. Ask three questions about any catalog being pitched as a stand-in for CMP. Can it govern a request that touches other systems, or does it only surface what that one tool can already do on its own? Does its access control work independently, or does a user still need separate credentials in the underlying tool just to see anything? Does it keep tracking cost and ownership after the job finishes, or does its job end the moment the template, module, or workflow completes? If the answer to all three is the narrower one, that catalog feature is only an improvement to its own tool. It is not a governance layer, no matter how the sales pitch frames it.

— THE BOTTOM LINE

Ansible, Terraform, and Backstage are not competing with CloudBolt CMP, and no organization needs all three running to get value from CMP.

Some teams will run all four layers, some will run two, some will run just one of the three alongside CMP. What stays constant is the layer that governs whatever combination is actually in place: approvals, cost, and ownership, whether the execution underneath is Ansible or Puppet, Terraform or a cloud console, Backstage or nothing at all. That is the knowledge this series set out to leave behind: not a fixed four-tool stack every organization needs to adopt, but a governance layer that belongs in the conversation no matter which of the other three, if any, are actually in the room.

[Talk to CloudBolt](#)

cloudbolt.io/demo