

How Spacelift's flexibility lets Lansweeper run IaC their way



Lansweeper

100%

Migration from Terraform to OpenTofu is complete (~60% Terragrunt; ~40% plain OpenTofu).

24,731

Resources managed as IaC have risen from ~6,000 to 24,731.

Summary

Lansweeper's platform team supports developers across both an on-premises product and a cloud product, with a strict GitOps philosophy and an opinionated Terragrunt setup. When Terraform Cloud's pricing changes, support quality, and post-acquisition trajectory stopped working for them, they evaluated env zero, Atlantis, and Atmos, and even considered building their own platform. Spacelift won on flexibility: policies, dependency orchestration, and a platform that fits the team's existing workflow rather than dictating a new one.

AI cyber asset intelligence platform Lansweeper provides a unified view of all your technology assets in one centralized solution. They ship both an on-premises product and a cloud product, and behind both sits a Kubernetes-native, cloud-native infrastructure that the platform team is responsible for keeping reliable, fast, and developer-friendly.

We spoke to Platform Team Lead Ismael Piñeiro Ramos and Senior Platform Engineer Juan Vela about why they moved away from Terraform Cloud, why they chose Spacelift, and how the platform fits their GitOps-first way of working.

The challenge for Lansweeper

Lansweeper's platform team was already on Terraform Cloud when several things stopped working at once. Support response slowed. The pricing model shifted toward a resource-based count. After the IBM acquisition, the team didn't believe things were going to improve, and the tool wasn't flexible enough to support the automation they wanted to build next.

"I opened several tickets to them requesting improvements, and they were not even answering most of the time," says Juan, Senior Platform Engineer at Lansweeper.

Platform Team Lead Ismael adds commercial and strategic context: "Once Terraform Cloud changed the pricing model to one based on resources, we started seeking other solutions. After the acquisition by IBM, we took the decision because we thought things were not going to improve. We were also working on some projects that required us to be more flexible with automation."

The team wanted to migrate to Terragrunt and build a one-click, PR-driven deployment workflow on top of an opinionated app of their own, but Terraform Cloud couldn't get them there.

Why Lansweeper chose Spacelift

The team evaluated env zero, Atlantis, and Atmos, and they also considered building their own platform. Self-hosted options stumbled on role-based access control, which is hard to manage cleanly when developers also need self-service.

Spacelift won on flexibility. Three capabilities mattered most: policies, dependency orchestration through stack DAGs, and a model close enough to Terraform Cloud that the team didn't have to rethink how they work with IaC.

"We found it has some very cool features like the policies — that was one of the things we found very useful for some of our use cases. Dependencies — how we can manage dependencies and orchestrate the DAG for our stacks — that was also a must for us because in the end we need to be able to orchestrate our stacks from the beginning in a way that everything is deployed without any human intervention," says Ismael.

Lansweeper's Spacelift experience

Lansweeper runs an opinionated, GitOps-first workflow. Developers don't click anything to provision infrastructure. They open a pull request against a central Terraform repository, and folder structure drives the rest.

That folder structure encodes regions and environments, which determines which Spacelift Contexts apply. Admin stacks create the Spacelift stack automatically. A Terraform module injected by the platform team exposes environment metadata so developers can enforce naming conventions and other guardrails. If a custom internal module exists for the use case, developers use it. If not, they write plain Terraform.

Juan explains the process from a developer's perspective: "They create a pull request to this specific GitHub repository and push the Terraform code for the infrastructure they want to deploy. It is very important that they check the plan in the CI attached to the PR. We use CI a lot, and the prediction of what's going to be deployed is very useful. Then we review and check for coherence, no collisions with other projects, that they're not doing anything weird. And then when they merge, they check in Spacelift — probably through GitHub looking at the CI — and then jump to Spacelift to see what's been deployed, how the apply went, and so on," says Juan.

That model maps onto a long list of Spacelift capabilities the team relies on day to day: stacks, contexts, policies, dependencies, Spaces for hierarchical permissions, the private registry for Terraform modules and providers, cloud integrations with OIDC so no static credentials ever touch the platform, and self-hosted worker pools for deploying into private networks. The team is also mid-migration from Terraform to OpenTofu, with Terragrunt running the core infrastructure the platform team owns directly. A Terragrunt wrapper handles bootstrapping in one click.

What they explicitly don't use is just as informative. Blueprints don't fit their GitOps philosophy. Intent and AI-driven provisioning don't either, because everything deployed must first land in the main branch of a repository.

"We are using AI in the company, and we are increasing AI usage, but always the AI needs to write the code and then push the code to our repository. From there we can trigger our processes to create the infrastructure. But not directly with MCP, using natural language to create infrastructure. Because we have processes to ensure we can recover our product from a disaster — basically, recreating everything from the repository. If we lose this history and information, we cannot recover anything in the same state," says Juan.

Spacelift's impact on Lansweeper

Two themes run through how the team talks about Spacelift: flexibility, and the kind of reliability you stop noticing.

"You don't need to adapt your way of working to Spacelift. You can adapt Spacelift to the way you work. And that's really good," says Ismael. "After adopting Spacelift we changed a lot of things in our way of working because we adopted a program, and thanks to Spacelift, we were able to make changes that were not possible before with Terraform Cloud."

On reliability, Ismael is matter-of-fact: "It just works. In general, the product works like it should and gives us what we need."

Juan goes further, naming reliability as a value the team rarely articulates because they assume it: "It's very reliable. Reliability and resilience are something we value a lot and just expect — and it's not something everybody does," says Juan.

The third change is the vendor relationship itself. Juan is one of Spacelift's most active customers on product feedback, and the contrast with the previous tool is sharp.

"It's night and day. The difference is huge. Not only because I get answers — which is what I think should happen — but also I see real changes behind the feedback I provide, when it makes sense ... I feel that we are heard and being taken care of. And I see that some new features are a result of our feedback," says Juan.

Asked what he'd tell a platform team weighing up alternatives, Juan keeps it simple: "If you want more freedom, flexibility, and an evolving platform — pick Spacelift."

This could be your story

Empower your platform team with Spacelift.

Liftoff with Spacelift!

