


# Spacelift as a Self-Service Powers Infrastructure Automation for Developers at commercetools

 spacelift

 commercetools

## >1 day

Before Spacelift, SREs spent at least a day per week attending to PRs and follow-ups.

## A few minutes

Since adopting Spacelift, SREs spend mere minutes on PRs.

### Summary

commercetools, the leading digital commerce platform for global enterprises, relies on infrastructure automation and deployment for its SRE organizational processes. SRE teams previously struggled to manage Terraform across complex multicloud environments, but Spacelift has freed them to focus on work that adds real business value while developers safely self-serve the infrastructure they need.

### About commercetools

commercetools enables companies to flexibly customize, tailor, and scale shopping experiences across markets with its headless, API-first, multi-tenant SaaS commerce platform. Infrastructure automation and deployment are fundamental to commercetools' SRE organizational processes. However, the SRE teams had to grapple with the challenges of managing Terraform modules and applying Terraform configuration across multicloud environments.

We spoke to commercetools Sr. SRE Chuka Durugo about how Spacelift helped commercetools transform its cumbersome Terraform management approach into a streamlined infrastructure automation engine for developers.

### The challenge Spacelift helped commercetools solve

Before adopting the Spacelift platform, commercetools operated a labor-intensive module release workflow. This is what it involved:

- A dedicated GitHub repository with sub-directories per module
- GitHub tags and releases for module versioning
- Updates to every module reference across the codebase for every module update

And these are the stages of the Terraform apply workflow:

1. A developer opens a GitHub Pull Request (PR).
2. Terraform plan is run locally and added to the PR description.
3. This gets reviewed and merged.
4. Code is applied locally to targeted environments.

Not surprisingly, an SRE could spend an entire workday attending to developers' PR requests, and this effort could even spill over to the next day because of apply failures, new fixes, and retries. Such an intensely manual delivery process was hugely cumbersome, inefficient, and ultimately unsustainable for the SRE teams. "Delivery timelines for developers were not guaranteed because every opened PR fell into a queue to be actioned by an SRE, and, unless it was urgent, this work usually took days," recalls Chuka.

The process may have achieved the desired end result, but a far simpler solution was required to accelerate module releases, leverage module versioning, and automate infrastructure deployments. The teams needed a way to alleviate the pain of manual intervention and give developer teams the autonomy to manage their releases themselves.

### Why commercetools chose Spacelift

Spacelift was not the only IaC management solution that commercetools considered. They also evaluated Semantic Release and Terraform Enterprise:

1. **Semantic Release:** This version management tool automates a module's release workflow process. commercetools could use it to create tagged module versions in the repository where the modules are hosted, but Terraform does not appear to support the use of semantic version constraint on GitHub sources to automatically pick up patch fixes (e.g., version = "~> 1.0.0"). You can only do this via a module registry.
2. **Terraform Enterprise:** commercetools could have utilized Terraform Enterprise to deploy and manage Terraform resources and workflows, as well as support their private module registry; however, it proved too costly.

Chuka outlines the following criteria for the ideal commercetools solution:

- Effectively manage the deployment of commercetools' Terraform resources
- Provide support for a private module registry where commercetools could host its modules and leverage semantic versions
- Be easy to use, especially for developer teams
- Offer flexible customization to suit commercetools' deployment patterns
- Be cost-effective

### The Spacelift experience for commercetools

Spacelift works smoothly with commercetools' current Terraform IaC configuration. It manages:

- all the heavy lifting of maintaining state:
  - During stack creation, Spacelift enables state management for the stack, but it's flexible enough that the team can manage it themselves if desired. "We have configured Spacelift to manage state only for Spacelift-specific resources such as stacks and spaces."
- integrating with Github repositories:
  - Installing and configuring Spacelift for commercetools' Github environment was seamless and straightforward. "We use Spacelift policies to configure a push policy that triggers a Run only upon changes related to the stack's project root, but it ignores the rest."
  - Configuring Spacelift to add a Plan comment to every PR after a successful Run helps teams to preview changes within the Pull request without having to navigate away from Github.
- authenticating to AWS and Google Cloud:
  - One of the ways Spacelift offers to authenticate with the cloud is a private worker pool. "This is a really important feature that helped us easily authenticate with our AWS environment, as that is the most suitable for our setup. Considering we operate a multicloud setup for Google Cloud we leverage OIDC for authentication."
- planning and deploying the Terraform configurations:
  - For every pull request or commit made by developer teams, a proposed run is triggered, which comments a plan of the PR. Whenever the pull request is merged, a tracked run is triggered, and the changes are deployed once the PR author approves and confirms them. "We use approval policies to make sure that a stack won't progress to the next phase without at least an approval from a relevant team."
- enabling collaborations between teams in the workflow process
  - "We used the approval policy to configure Spacelift stacks so that a tracked run created when a pull request author merges their PR is applied only when it is approved by the team that owns the space where the stack was created. The teams are now confident that someone outside of their team cannot make any infra change without their knowledge or approval."

Spacelift also comes with a private module registry, which neatly solves the problem of module versioning and accelerates releases across all its references. Teams can now leverage the Terraform versioning syntax "version = ~> 1.0.0", and they no longer have to manually update all references after a module patch version update.

### Spacelift's impact on commercetools

With Spacelift, delivery timelines for developer teams have been slashed. Deployment times that previously extended into days have been reduced to mere minutes. Chuka explains that this is largely because teams no longer depend on SRE teams for deployment: "What contributed to the long wait time was the dependency on SRE availability. With Spacelift, that dependency is no longer there, thereby reducing deployment time to minutes." Developers have the autonomy to manage their own deployments. They simply get their PRs merged to trigger the stack workflow, and deployment happens as soon as the relevant team approves.

“

SRE teams can now focus their time on more meaningful work because developer teams are empowered to self-manage their deployments via Spacelift.

**Chuka Durugo**

Senior Site Reliability Engineer at commercetools

## This could be your story

Empower your platform team with Spacelift.

[Liftoff with Spacelift!](#)