

Ping Identity modernizes its Terraform at scale with Spacelift

spacelift

PingIdentity.

30 minutes

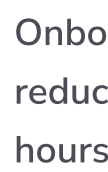
Time saved per deployment from automated plan generation and inline MR summaries

3X

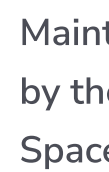
Resources managed as IaC have risen from ~6,000 to 24,731.

Summary

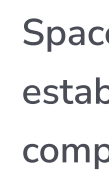
Ping Identity, a leader in securing digital identities for the world's largest enterprises, is modernizing how it manages Terraform at scale. Previously reliant on aging, bespoke Jenkins pipelines written in Groovy, Ping's SRE teams faced growing bottlenecks, limited policy enforcement, and increasing maintenance overhead. Spacelift is now part of Ping Identity's approach to migrating critical workloads off Jenkins and into a scalable, policy-driven Terraform workflow



Onboarding a new Terraform stack has reduced from weeks with Jenkins to mere hours using Spacelift's Terraform provider configuration.



Maintenance overhead has been slashed by the shift to managing stacks via Spacelift.



Spacelift has helped Ping Identity establish a foundation for policy-driven compliance, cost control, and blast radius management.

Ping Identity is a leader in securing digital identities for the world's largest enterprises. With over 2,200 employees and 750+ engineers, its engineering organization supports complex infrastructure across AWS and Cloudflare using Terraform as its primary infrastructure-as-code tool.

The APAC SRE team is responsible for managing and modernizing significant portions of this infrastructure. To support rapid scaling and enhance an already robust compliance framework, Ping Identity sought to transition from custom, maintenance-heavy pipelines to a highly automated architecture.

The challenge for Ping Identity

Legacy Jenkins pipelines were a bottleneck

Ping Identity's Terraform workflows were "wrapped" in highly customized Jenkins pipelines written in Groovy. Over time, this approach became increasingly difficult to maintain because the pipelines contained bespoke libraries and modified internal components, which were difficult to maintain due to the departure of many of their original authors and a lack of documentation and Groovy-trained engineers. Furthermore, Jenkins itself was aging and difficult to upgrade.

As Senior SRE Tim Hugall explains, "the codebase for our Jenkins pipelines was not well-understood. It was inefficient and excessively burdensome to maintain, let alone expand." Onboarding a new Terraform stack into Jenkins could take weeks of pipeline development, slowing down delivery and discouraging teams from expanding infrastructure as code.

Scaling Terraform across large, complex stacks

Ping Identity operates some large Terraform stacks (particularly around API Gateway) with tens of thousands of resources and deployments that can take hours to complete.

The team needed to streamline deployments and create a foundation for multi-account AWS strategies with more complex dependency management and enforced compliance and policy guardrails at scale.

Limited global policy enforcement

Under Jenkins, enforcing universal compliance, tagging, cost, or blast-radius policies across infrastructure deployments was difficult.

Ping Identity needed a policy engine deeply integrated into the Terraform workflow so they could enforce mandatory tagging, implement blast radius checks on destructive changes, control costs, and balance speed with governance.

Why Ping Identity chose Spacelift

Spacelift came to Ping Identity through internal recommendations and a prior proof of concept run by another SRE team. After further evaluation, Spacelift stood out for the following reasons:

- Particularly impressive policy capabilities for compliance, security, cost control, etc.
- Ability to self-host workers and VCS agents while using the SaaS control plane met internal security requirements for private GitLab integration
- First-class support for private GitLab instances via self-hosted VCS agents, with excellent merge request integration including plan summaries and conditional merging
- Powerful, easy-to-use dependency management functionality capable of intelligent interpretation of complex stack dependencies
- A well-documented and reliable Terraform provider enabling all Spacelift resources to be managed as code
- Possibilities afforded by the module registry
- Potential to use Terragrunt
- More intuitive than alternatives, reducing the learning curve for team members without sacrificing power or flexibility

Ping Identity's Spacelift experience

The APAC team took an incremental approach to its Spacelift migration, beginning with smaller, less complex stacks (such as Lambda authorizers and Cloudflare configurations). This allowed them to demonstrate immediate value to the business, replace manual plan generation with automated MR plan summaries, and show measurable engineering productivity gains.

Automated plan generation directly in GitLab merge requests has already saved multiple engineering hours per month because engineers no longer have to generate and paste plans into MRs.

An example of Ping Identity's repo set-up:

Path	Terraform providers used	Purpose
sre/terraform/spacelift	Spacelift	Contains Terraform, which controls the basic, shared Spacelift resources, such as spaces, VCS agent, worker pool, etc.
sre/terraform/modules/spacelift	Spacelift	Subgroup for Spacelift-specific modules including their 'stack' module
sre/terraform/authorizer	AWS	Basic example of Terraform managed/deployed by Spacelift. Used to be in a monorepo.
sre/terraform/cloudflare/cloudflare-pingone/spacelift	CloudFlare & Spacelift	More complex example of Terraform managed/deployed by Spacelift. Uses administrative stacks which deploy other Spacelift stacks which are used to deploy the CloudFlare stacks.

Streamlined Terraform-native workflows

Eliminating custom Groovy-based Jenkins logic with Spacelift means that stacks are configured using Terraform (via the Spacelift provider), all Spacelift resources are managed as code, and engineers work within familiar Terraform patterns.

"The shift from maintaining custom Groovy pipelines to managing stacks via Terraform, which all team members already know, has significantly reduced maintenance overhead."

Advanced configuration management with contexts

Ping Identity uses contexts with attached variables, mounted files and hooks, and they also use the autoattach label functionality. In one notable example, the team managed hierarchical configuration using Spacelift contexts when they decided to bring the Terraform code that controls their CloudFlare resources under Spacelift management.

Previously, their CloudFlare Terraform code had used the Hiera Terraform provider plus remote state references to implement a form of hierarchical configuration. The hierarchical configuration parameters themselves were stored in YAML files inside a hierarchical directory structure and committed. The excerpt below demonstrates the hierarchy:

From modules/context/main.tf:

```
data/test
├── common_tier.yaml
├── usa
│   ├── common_geography.yaml
│   ├── example-one-a.com
│   │   ├── common_cdn_zone_records_acm.json
│   │   ├── common_cdn_zone_records.yaml
│   │   ├── common_cdn_zone_transform_rules.yaml
│   │   └── common_cdn_zone.yaml
│   ├── example-one-b.com
│   │   ├── common_cdn_zone_records_acm.json
│   │   ├── common_cdn_zone_records.yaml
│   │   └── common_cdn_zone.yaml
│   ├── example-b.com
│   │   └── common_cdn_zone.yaml
│   └── example-two-a.com
│       ├── common_cdn_zone_records_acm.json
│       ├── common_cdn_zone_records.yaml
│       ├── common_cdn_zone_transform_rules.yaml
│       └── common_cdn_zone.yaml
│   └── example-one-b.com
│       ├── common_cdn_zone_records_acm.json
│       ├── common_cdn_zone_records.yaml
│       └── common_cdn_zone_records.yaml
```

The YAML files and their directories were retained as the source of truth for the hierarchical configuration, but the configuration was implemented using Spacelift contexts. Here are excerpts from the Terraform code used to generate these contexts from the YAML files:

```
resource "spacelift_environment_variable" "json_encoded" {
  for_each = {
    for k, v in var.config_file_contents : k => jsonencode(v) if !can(tostring(v))
  }
  context_id = spacelift_context_this.id
  name       = "TF_VAR_${each.key}"
  value      = each.value
  write_only = false
}

resource "spacelift_environment_variable" "string" {
  for_each = {
    for k, v in var.config_file_contents : k => v if can(tostring(v))
  }
  context_id = spacelift_context_this.id
  name       = "TF_VAR_${each.key}"
  value      = each.value
  write_only = false
}
```

From root module:

```
module "cloudflare_base_context" {
  source = "../modules/context"
  name   = "cloudflare"
  stack_ids = flatten([
    for m in local.stack_ids : m.stack_ids
  ])
  config_file_contents = merge(
    yamldecode(file("../data/transform_rules.yaml")),
    yamldecode(file("../data/common.yaml"))
  )
  priority = 5
}

module "cloudflare_test_context" {
  for_each = toset(["test", "staging", "prod"])
  source   = "../modules/context"
  name     = "cloudflare:${each.value}"
  stack_ids = flatten([
    for m in local.stack_ids : m.stack_ids if m.tier == each.value
  ])
  config_file_contents =
    yamldecode(file("../data/${each.value}/common_tier.yaml"))
  priority = 4
}
```

Spacelift's impact on Ping Identity

Although the migration is ongoing, Ping has already accelerated stack onboarding, increased deployment frequency, and reduced operational complexity. The company is building a foundation for multi-account and large-scale modernization, using Spacelift to dismantle massive API Gateway stacks, implement stronger dependency management, and enforce universal governance policies. The long-term goal is to migrate all Jenkins-based Terraform deployments to Spacelift, with API Gateway stacks as the top priority. Ping Identity views Spacelift as the platform that will allow them to modernize and scale infrastructure safely.

As Tim summarizes:

"It's good to have pipeline software specifically designed with Terraform in mind. Spacelift is easier to work with than bespoke scripts or general CI tools. Everything's been thought of."

This could be your story

Empower your platform team with Spacelift.

Liftoff with Spacelift!