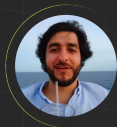



LaunchDarkly →

# Building Dynamic Configuration into Terraform




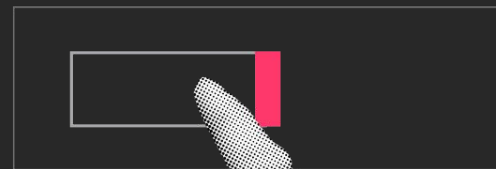
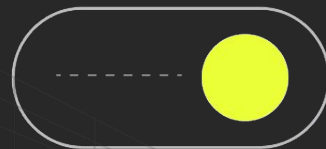
Hosh Sadiq  
Platform Engineer

 @hoshsadiq



Isabelle Miller  
Software Engineer

 @sloloris



# Who We Are / About Us



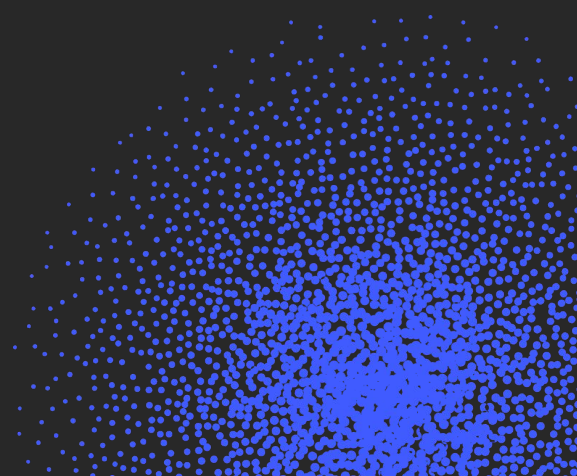
**Hosh**

- ✓ Platform Engineer at LaunchDarkly
- ✓ Really bomb at Terraform



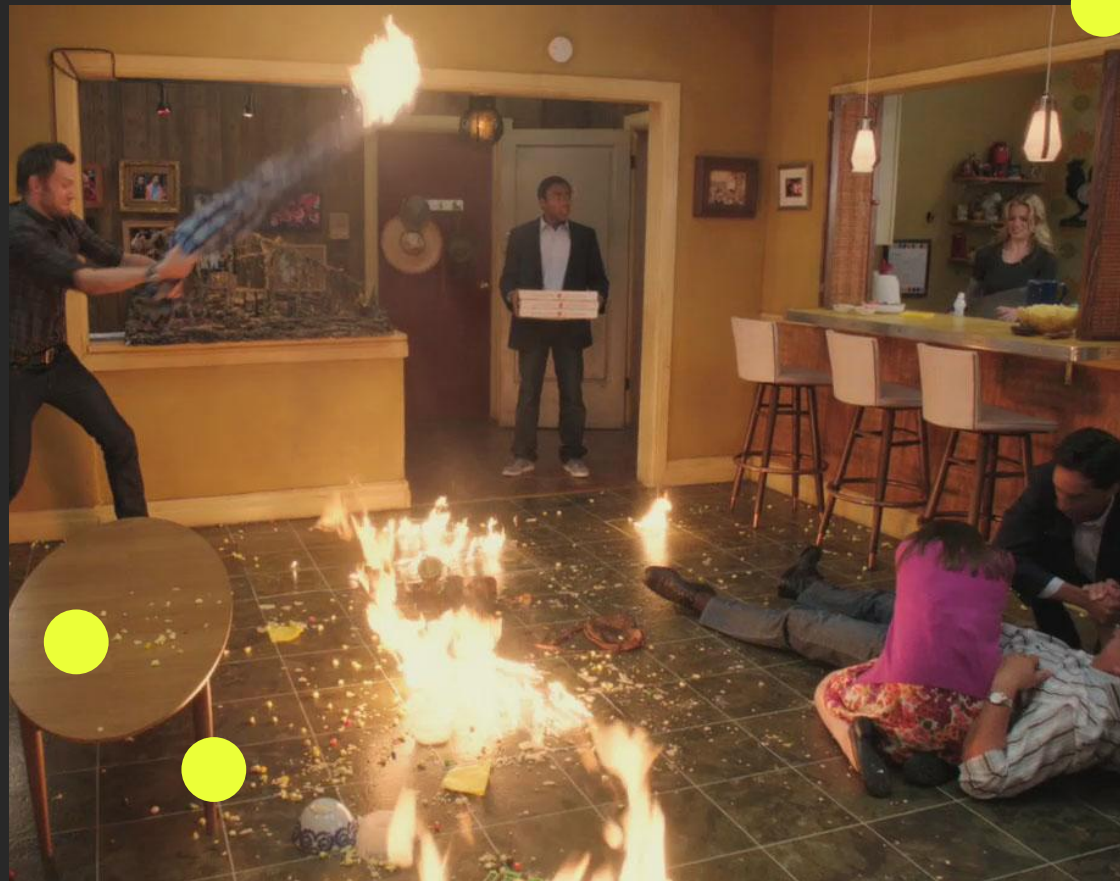
**Isabelle**

- ✓ Works on integrations at LaunchDarkly
- ✓ Maintainer of the LD Terraform provider



# Startups, eh?

Scaling is hard



# Terraform management at LaunchDarkly

Tl;dr oops monorepo

- 🚀 Over 100 software engineers for one tiny platform team
- 🚀 You build it, you run it
- 🚀 Almost all teams touch or maintain some Terraform configuration
- 🚀 Migrating Terraform is a pain



## When in Rome...



\* Ask us about how we use flags to do rate limits, pricing tiers, and more.

\*\* This is an unreleased feature. Please do not complain to sales that you cannot find it yet.



# Terraform: A New Provider

```
data "ldflags_evaluation_int" "k8s_replicas" {
  count      = 2
  flag_key   = "myflag"
  default_value = 1

  context = {
    key = "myapp-${count.index}"
  }
}


resource "kubernetes_deployment" "ldflags_app" {
  count = 2

  // ...
  spec {
    replicas =
    data.ldflags_evaluation_int.k8s_replicas[count.index].value
    // ...
  }
}
```

\* We're both also Terraform nerds, so it was a great opportunity to test out Terraform's experimental new SDK, which features actual support for nested attributes, custom types, and detecting unknown or null values. Ask us about the tradeoffs!

# Why feature management in Terraform?

## Key Takeaways



### Avoid small code changes

No need to go through a pesky PR process to make small and harmless changes like changing a monitoring or SLO threshold.

### Controlled access

Your feature management platform can further limit who can make changes.

### More independence

No need to involve Platform/DevOps teams for config updates, and engineers don't need to re-learn Terraform every time they dip their toes.

### And more...?





```
data "ldflags_evaluation_json" "slo_availability_thresholds" {
  flag_key      = "slo-availability-thresholds"
  default_value = jsonencode({
    target = 99.9
    warning = 99.95
  })
  context = {
    key = "webhooks"
  }
}
```

```
module "slo_availability" {
  source      = "./modules/slo"
  environment = "production"
  thresholds  = jsondecode(data.ldflags_evaluation_json.
    slo_availability_thresholds.value)
}
```

[Feature flags](#) / SLO Availability Thresholds

## SLO Availability Thresholds

Maintained by  Isabelle Miller

Created 2 months ago

slo-availability-thresholds

Percentage of successful requests

Targeting

Workflows

Insights

Experimentation

Variations

History

Code references

Links

Settings


### Targeting



Review and save 

On

 Evaluated 2 weeks ago

 Currently serving Tier 1 - Target 99.99, Warning 99.995, Tier 2 - Target 99.95, Warning 99.99

#### Prerequisites

+ Add prerequisites

#### Individual targeting

+ Add user targets

#### Target users who match these rules

Tier 1 Availability



key




is one of



webhooks 



SERVE

 Tier 1 - Target 99.99, Warning 99.995


Add rule

Default rule

SERVE

 Tier 1 - Target 99.99, Warning 99.995  
Highest Availability Tier

 Tier 2 - Target 99.95, Warning 99.99

 Defaults - Target 99.9, Warning 99.95  
Default Availability Thresholds


a percentage rollout

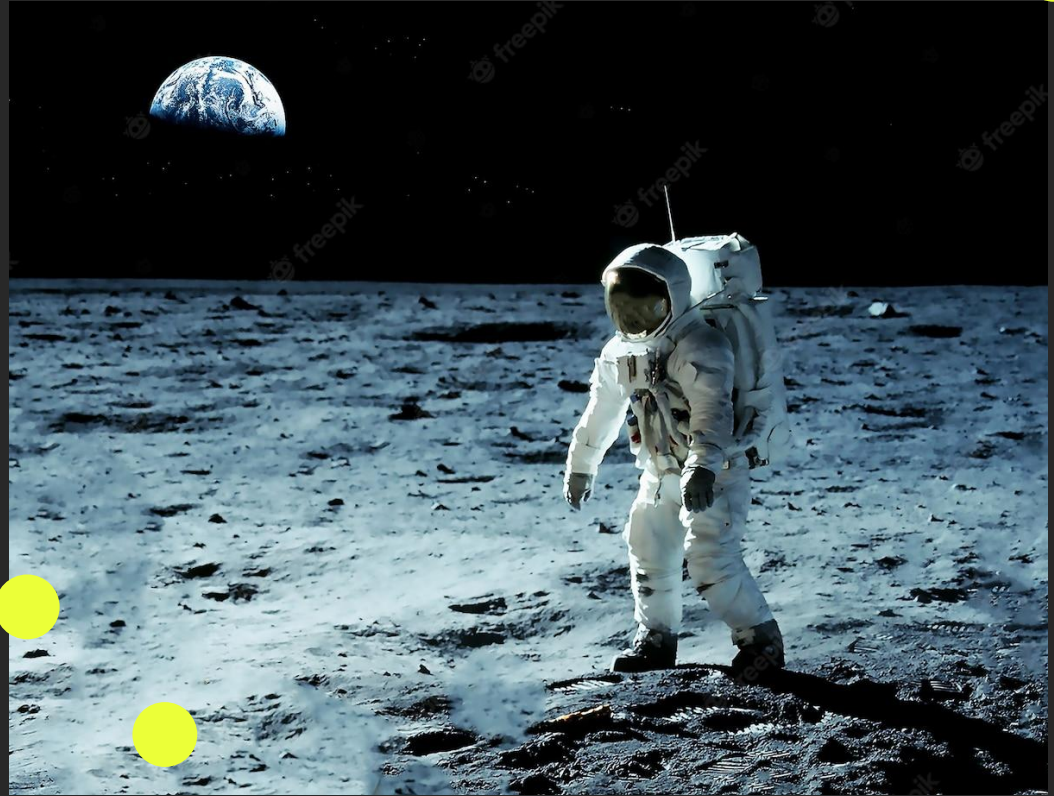
If targeting is off, serve  ...





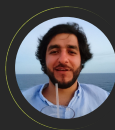
# A New Frontier

 @launchdarkly  
@sloloris



LaunchDarkly →

# That's all, folks!




Hosh Sadiq  
Platform Engineer

 @hoshsadiq



Isabelle Miller  
Software Engineer

 @sloloris

