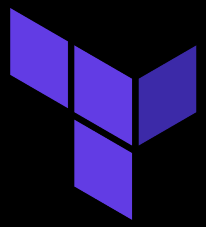


HashiCorp

Terraform

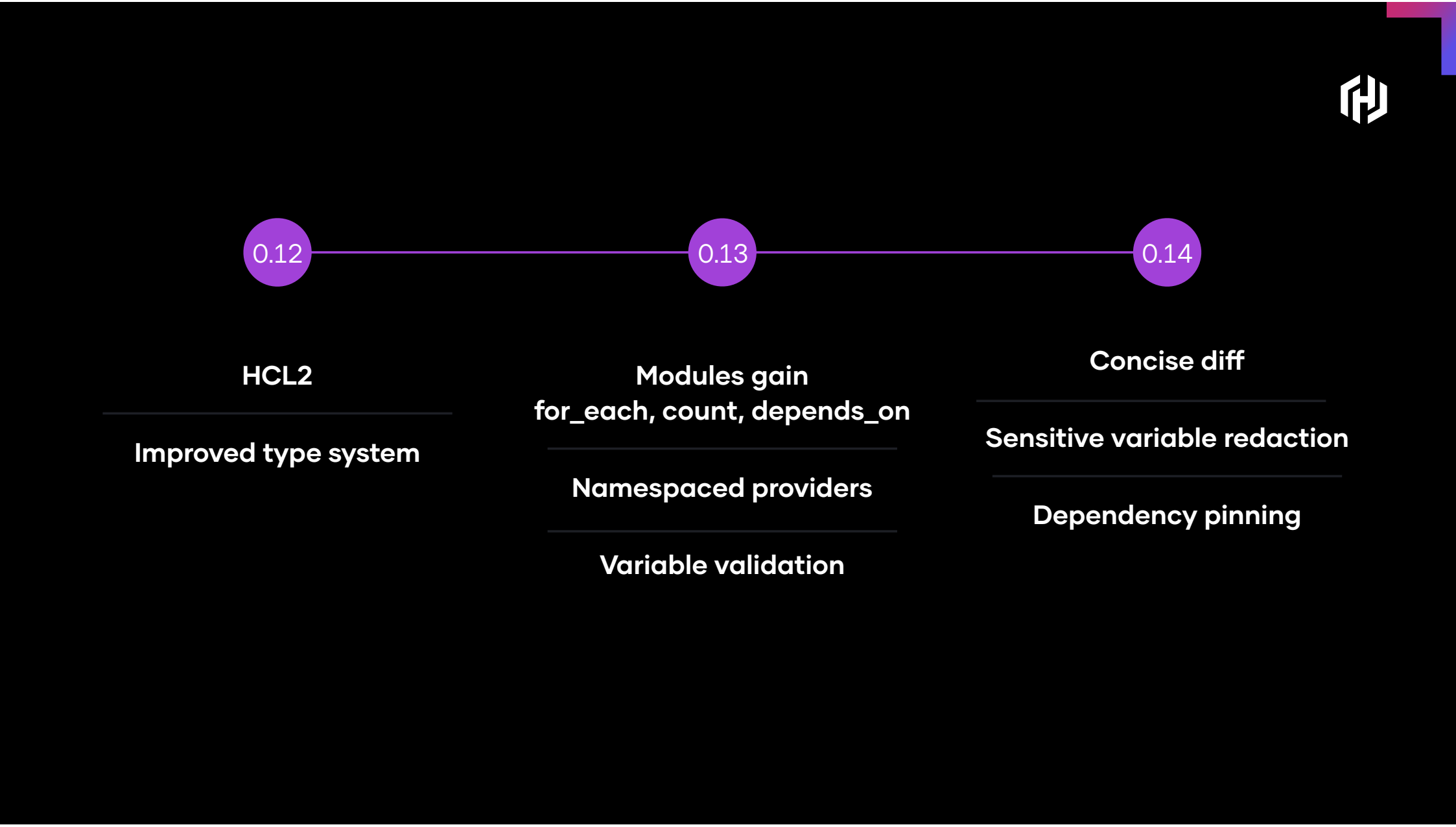
HashiCorp

From CLI to Cloud and Back!

Kyle Ruddy | Petros Kolyvas

HashiCorp
Terraform

Terraform CLI

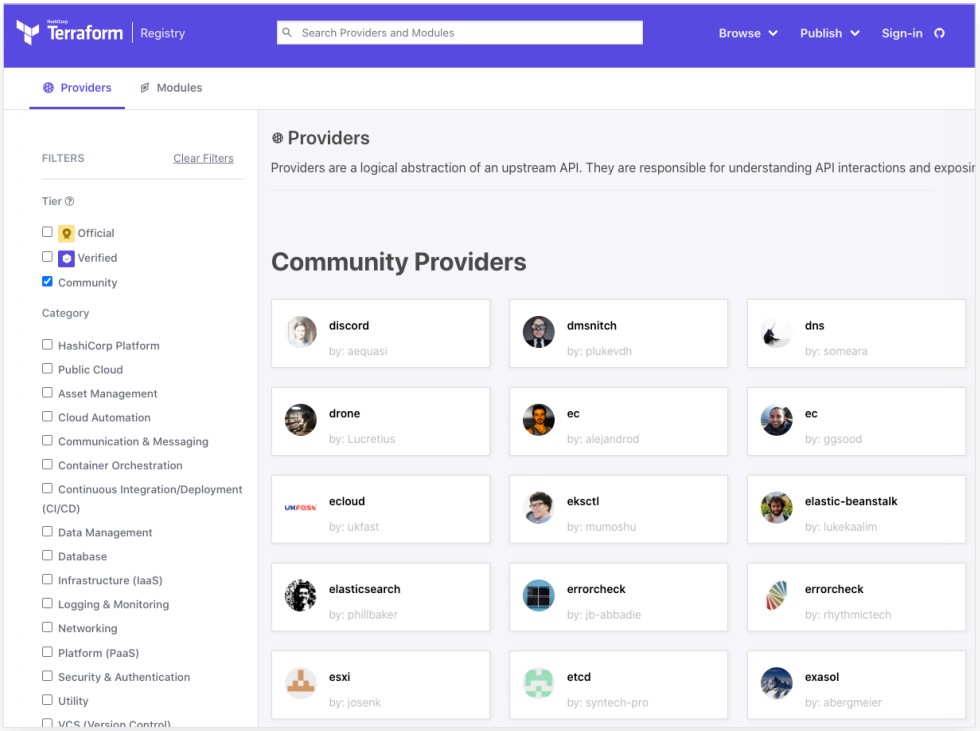

0.12
HCL2
Improved type system
0.13
Modules gain
for_each, count, depends_on
Namespaced providers
Variable validation
0.14
Concise diff
Sensitive variable redaction
Dependency pinning

HashiCorp
Terraform

Terraform 0.13

Community Providers

From 0-350+ in less than two months

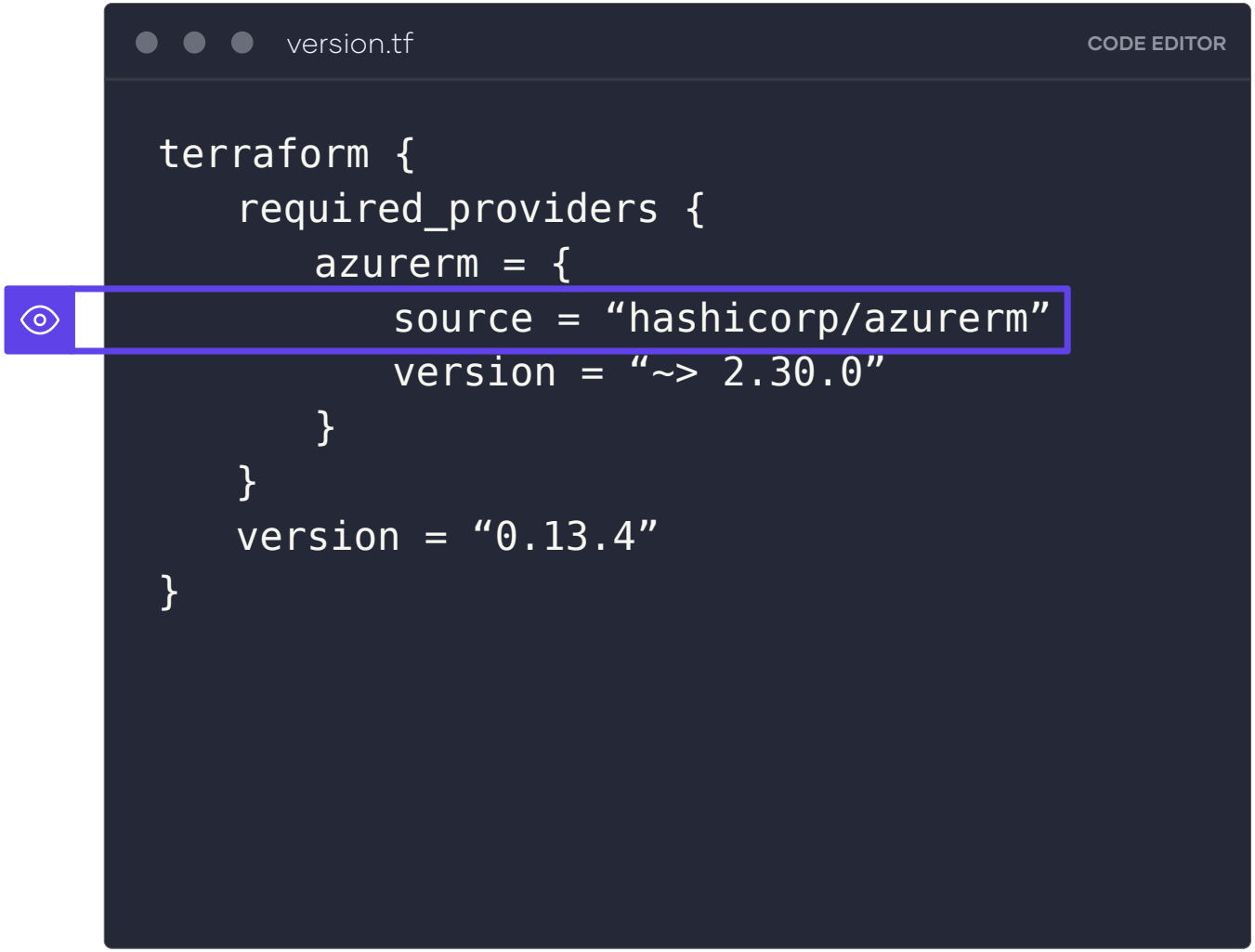


Community breeds variety

- ACME
- git (x4)
- shell
- stdlib
- wireguard

Provider Source

Understanding the source address



Provider Source

Understanding the source address

version.tf CODE EDITOR

```
terraform {
    required_providers {
        azurerm = {
            source = “hashicorp/azurerm”
            version = “~> 2.30.0”
        }
    }
}
```

Source address

`<registry_hostname>/<namespace>/<type>`

Inferred address without registry host

`registry.terraform.io/hashicorp/azurerm`

Provider Installation

File system mirrors and network mirrors.

Provider Installation

Network mirror pro-tip

Riot Games and 0.13

module for_each &
variable validation
FTW

Riot Games and 0.13

module for_each &

variable validation

FTW

```
data "local_file" "spaces" {
  for_each = fileset("./spaces_yaml", "*.yaml")
  filename = "./spaces_yaml/${each.value}"
}

locals {
  space_info = {
    for name, data in data.local_file.spaces :
    // This merge serves three purposes :
    //  1 - Inject the space name based on the filename
    //  2 - Inject ldap_path into every blob
    //  3 - Ensure all optional properties are preset
    replace(name, ".yaml", "") => merge({
      space_name : replace(name, ".yaml", ""),
      ldap_path : vault_ldap_auth_backend.ldap.path,
      allow_aws_iam_auth_from : [],
      aws_iam_read_only : {},
      aws_iam_read_write : {},
      approle_read_only : {},
      approle_read_write : {},
    }, yamldecode(data.content))
  }
}

module "spaces" {
  for_each = local.space_info
  source   = "./modules/space"

  space_info = local.space_info[each.key]
}
```

Riot Games and 0.13

module for_each &

variable validation

FTW

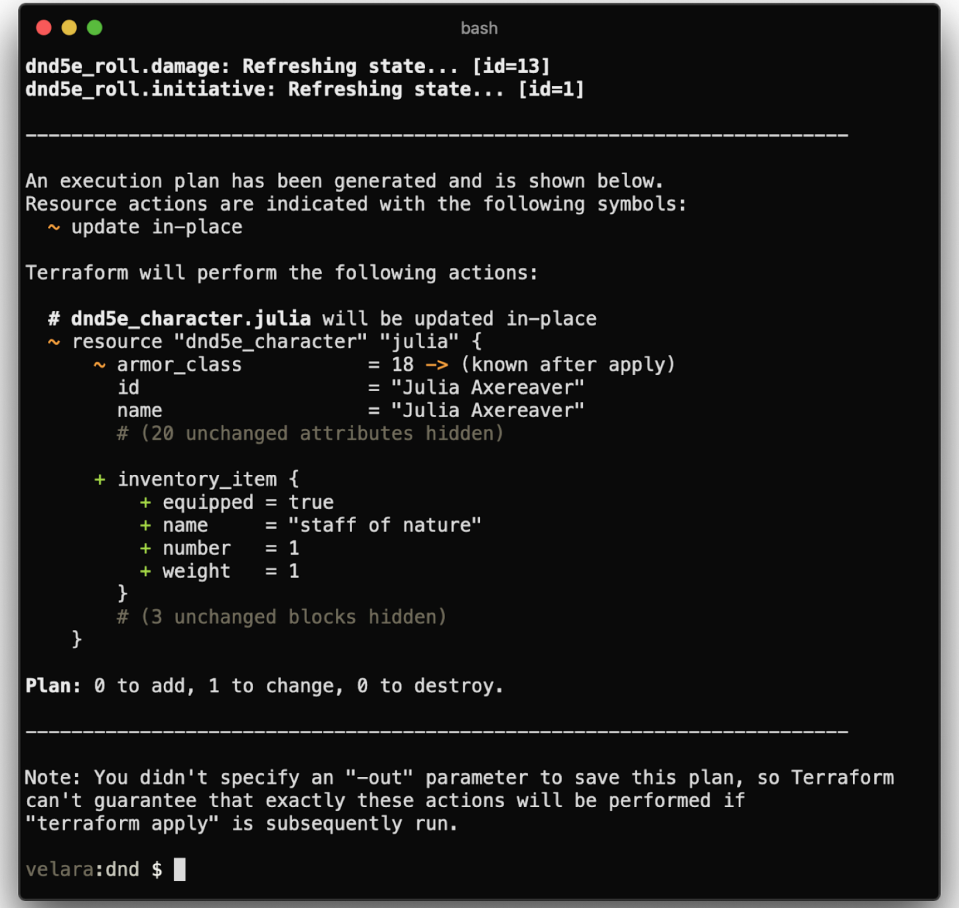
```
variable "space_info" {
  type = object({
    space_name : string,
    owners : set(string),
    ldap_path : string,
    allow_aws_iam_auth_from : set(string),
    aws_iam_read_only : map(object({
      arns : set(string),
      paths : set(string)
    })),
    aws_iam_read_write : map(object({
      arns : set(string),
      paths : set(string)
    })),
    approle_read_only : map(object({
      paths : set(string)
    })),
    approle_read_write : map(object({
      paths : set(string)
    }))
  })

  // Space name validation
  validation {
    condition     = can(regex("^[a-z_]*$", var.space_info["space_name"]))
    error_message = "The space name must match ^[a-z_]*$."
  }

  // Owner validations
  validation {
    condition     = length(var.space_info["owners"]) <= 3
    error_message = "A maximum of 3 owners is allowed."
  }

  validation {
    condition = ! contains([
      for owner in var.space_info["owners"] :
      can(regex(".*@riotgames\\.com", owner))
    ], false)
    error_message = "All owners must be an @riotgames.com email."
  }

  // Path validations
  validation {
    condition = ! contains(flatten([
      for value in values(var.space_info["approle_read_only"]) :
      value["paths"]
    ]), "")
```

HashiCorp
Terraform

Terraform 0.14

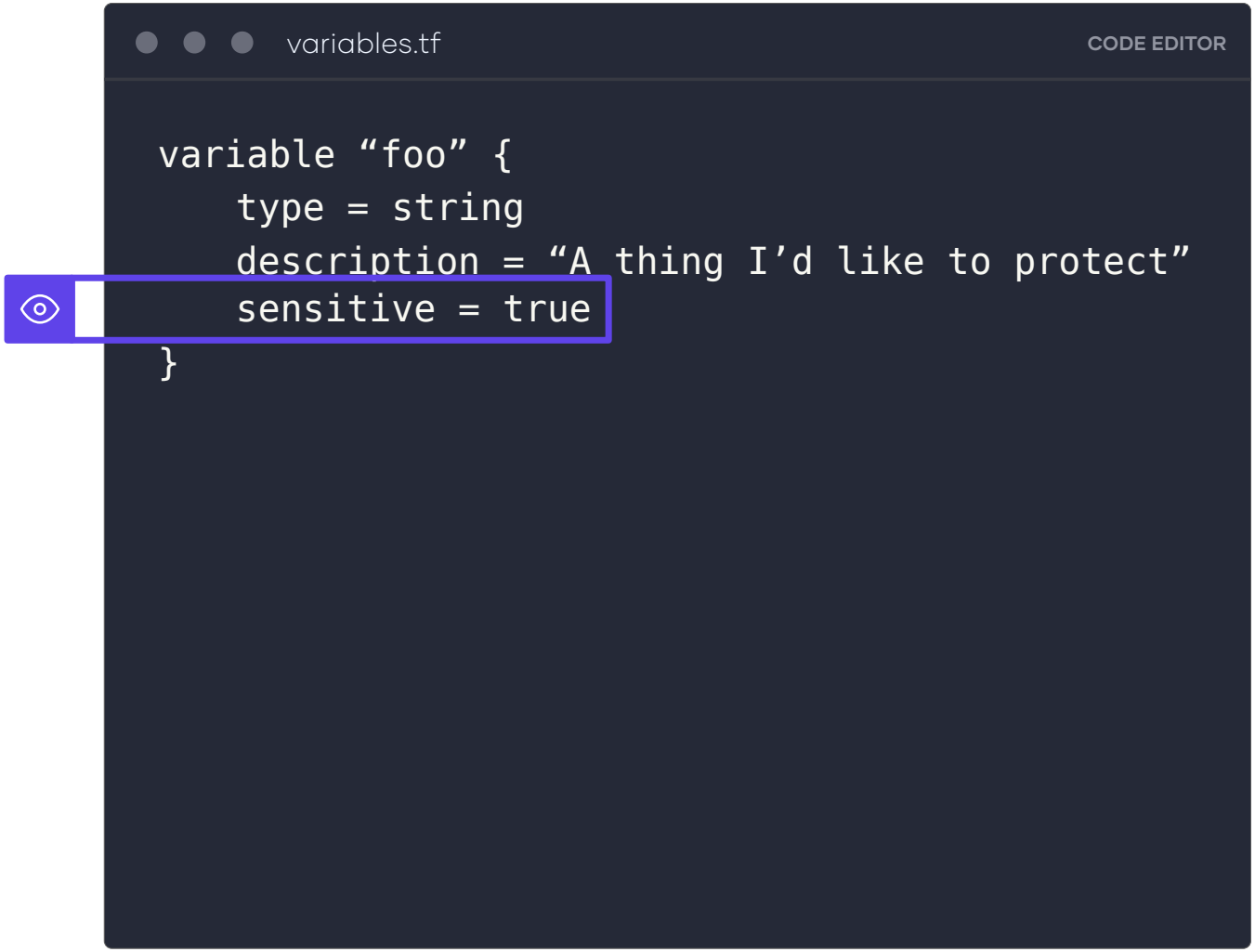
Concise Diff

Making Terraform's diffs more actionable



Sensitive values in Terraform

Protecting things with people you trust



Terraform Meta

Terraform testing

We're not at `>terraform test` just yet

test.tf — CODE EDITOR

```
terraform {
    required_providers {
        testing = {
            source = “apparentlymart/testing”
            version = “~> 0.0.2”
        }
    }
}
```

Provider-powered

Beginning our discovery work around a proof of concept from Terraform's tech lead, Martin Atkins.

HCL-centric

Write tests for your modules using data source-based assertions or the test anything protocol.

HashiCorp
Terraform

Demo

HashiCorp
Terraform

One more thing

linux_arm64

Try it today.