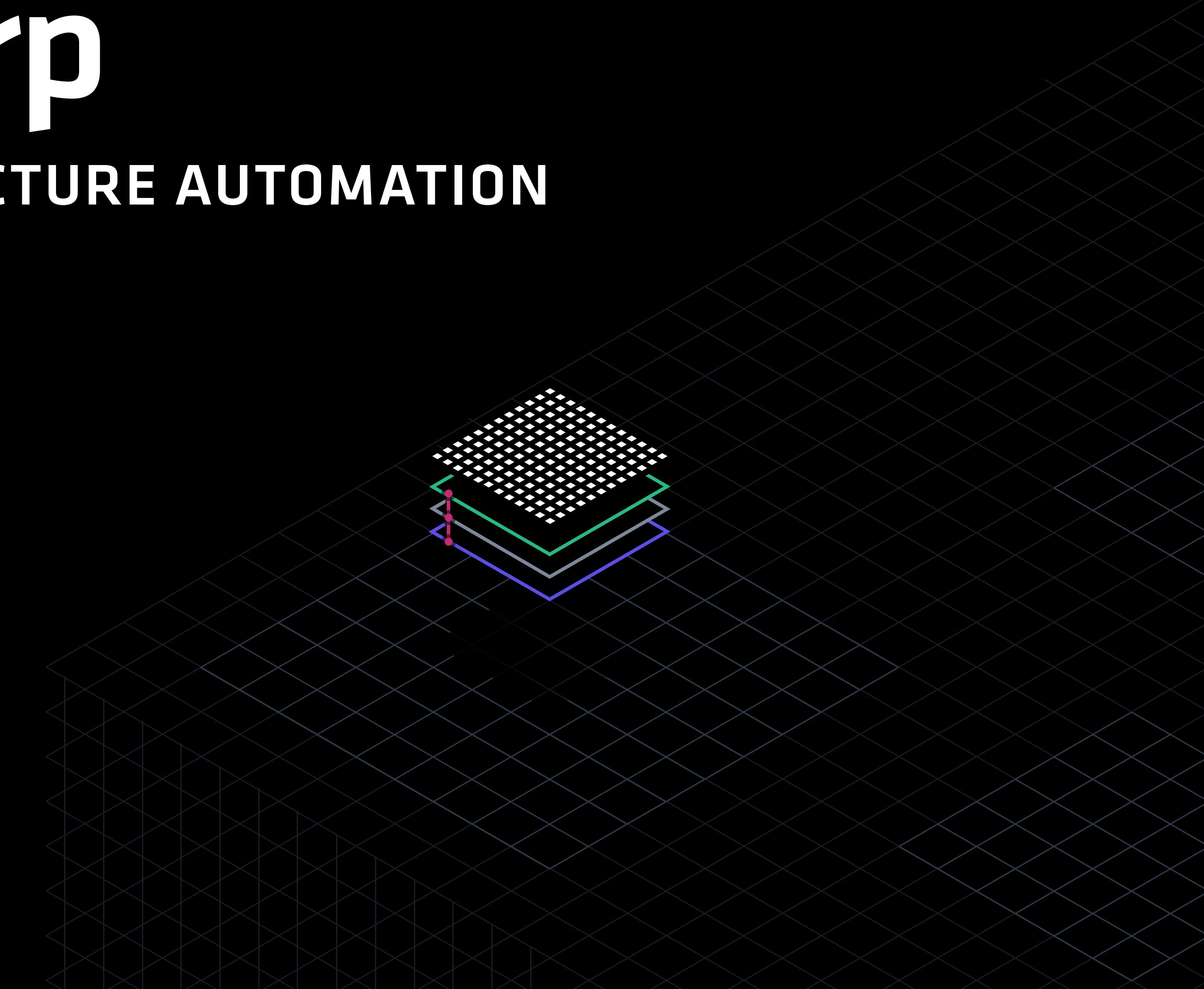




CLOUD INFRASTRUCTURE AUTOMATION

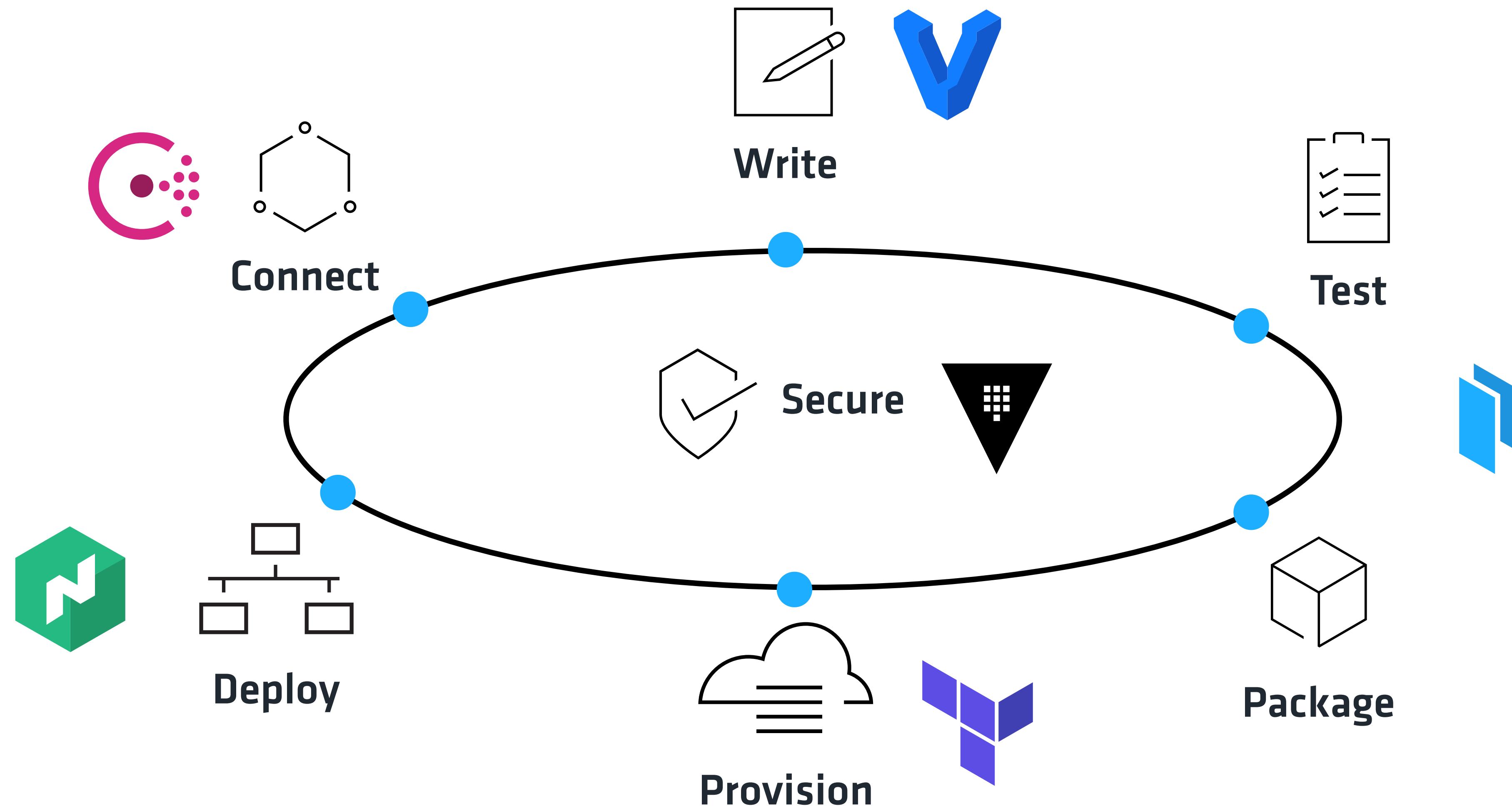




Terraform
ENTERPRISE

Introduction

Application Delivery Lifecycle



Agenda

Syntax Review

Remote State

Cloud State

Teams

TFE Basics

Collaboration

Useful Providers

Packer

Multi-Provider Configs

Lifecycles

Module Registry

Configuration Designer

We will cover

Tools for teams and scale

Hosted Terraform Enterprise

Examples use AWS

Q&A and your requests

We will not cover

Terraform Enterprise API

Private Terraform Enterprise

Source code management tools

Configuration management tools

Every cloud

What You'll Learn

- ✓ Use collaborative features of Terraform
- ✓ Organize Terraform projects for large scale infrastructure installations
- ✓ Use state
- ✓ Become confident with supporting tools

Review

Key Terraform Syntax & Features

- ✓ Summary of providers, resources, variables, outputs
- ✓ Summary of modules
- ✓ Tips for upgrading commands and providers

Terraform Syntax

Building infrastructure with Terraform

Providers - Cloud, VCS, Monitoring, etc.

Resources - EC2 Instances, Credentials, etc.

Variables - Input for tagging or defining resources.

Outputs - Used for record keeping or as a variable to pass into other modules.

Leveling up by utilizing modules

```
my-infrastructure
├── main.tf
├── output.tf
├── terraform.tfvars
├── terraform.tfstate
└── variables.tf
```

```
provider "aws" { }
```

TIP

Secure Credentials with ENV Variables or Cloud Role

For flexibility and to nearly eliminate the possibility of accidentally committing cloud credentials to source code control systems, consider storing cloud credentials in environment variables. Or use cloud role.

If they are named correctly, the appropriate Terraform provider will discover and use them.

```
export AWS_ACCESS_KEY_ID="AAAAAAA"  
export AWS_SECRET_ACCESS_KEY="XXXXXXXXXX"  
export AWS_DEFAULT_REGION="us-west-2"
```

```
$ source ~/.config/envs/aws
```

WARNING

Providers Must Be Explicitly Upgraded

We value stability and we respect your control over your environment.

By default, Terraform will lock on to the first version available when you first run `init` on a configuration.

To upgrade, you must explicitly state a version in your configuration, or run this command:

```
terraform init -upgrade
```

```
provider "aws" {  
  version = ">= 1.19.0"  
}
```

```
resource "aws_instance" "web" {  
    ami              = "ami-e5d9439a"  
    instance_type   = "t2.nano"  
}
```

```
variable "ami" {}

resource "aws_instance" "web" {
    ami          = "${var.ami}"
    instance_type = "t2.nano"
}
```

```
variable "ami" {}

resource "aws_instance" "web" {
    ami          = "${var.ami}"
    instance_type = "t2.nano"
}

output "public_ip" {
    value = ["${aws_instance.web.public_ip}"]
}
```

```
# ENVIRONMENT VARIABLES
# Define these secrets as environment variables

# AWS_ACCESS_KEY_ID
# AWS_SECRET_ACCESS_KEY

variable "images" {
    type = "map"
    default = {
        us-east-1 = "image-1234"
        us-west-2 = "image-4567"
    }
}
variable "zones" {
    type = "list"
    default = ["us-east-1a", "us-east-1b"]
}
```

```
vpc_module
└── main.tf
└── output.tf
└── variables.tf
```

```
● ● ●
```

```
module "vpc" {
  source = "terraform-aws-modules/vpc/aws"
}
```

```
module "vpc" {
  source = "terraform-aws-modules/vpc/aws"

  name      = "my-vpc"
  cidr     = "10.0.0.0/16"

  azs          = [ "eu-west-1a", "eu-west-1b" ]
  private_subnets = [ "10.0.1.0/24", "10.0.2.0/24" ]
  public_subnets  = [ "10.0.101.0/24", "10.0.102.0/24" ]

  enable_nat_gateway = true
  enable_vpn_gateway = true

  tags = {
    Terraform = "true"
    Environment = "dev"
  }
}
```

```
# Useful built-in functions
basename(path)
cidrhost(iprange, hostnum)
concat(list1, list2, ...)
file(path)
join(delim, list)

my-list[ 7 ]
```

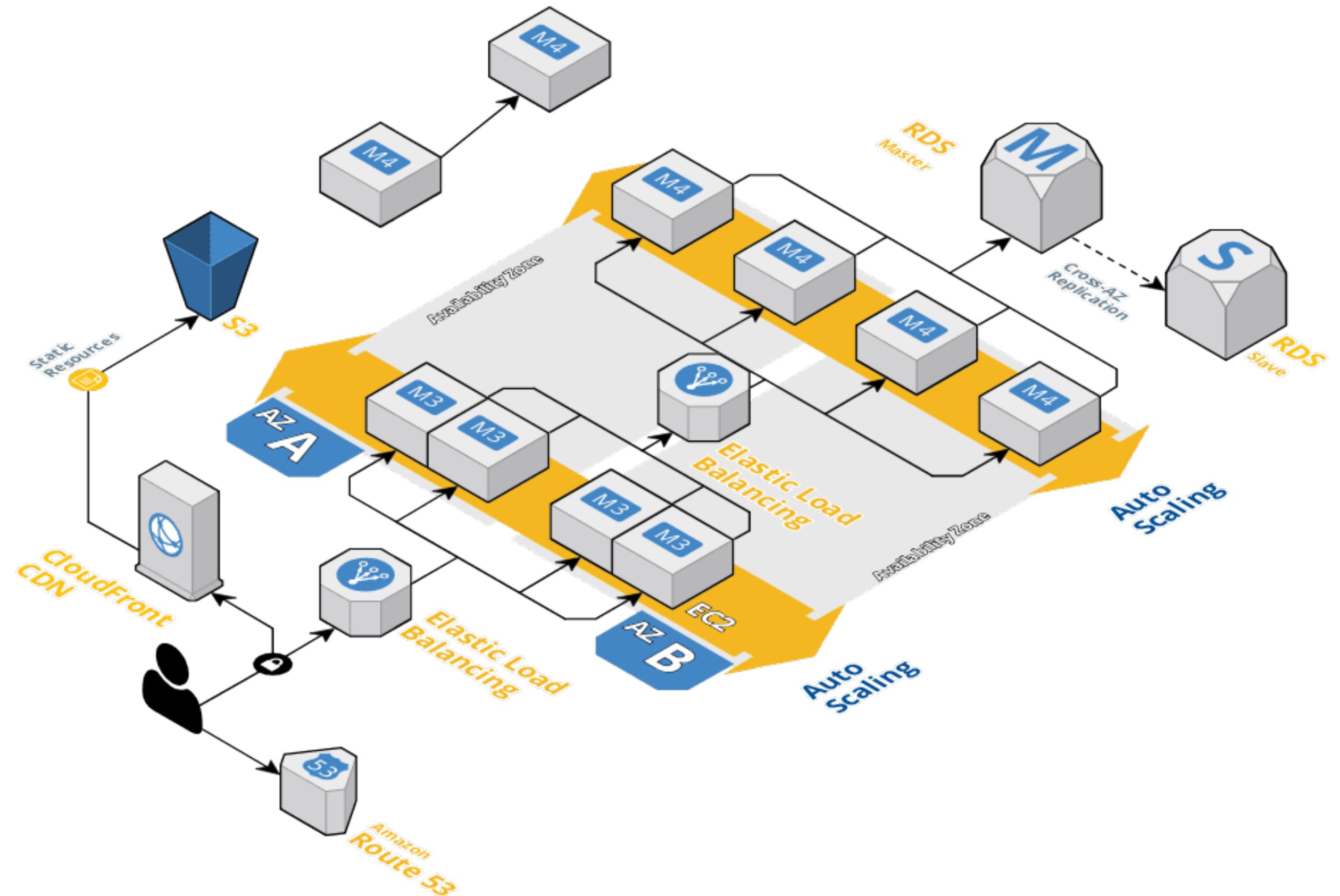
Key Terraform Syntax & Features

- ✓ Summary of providers, resources, variables, outputs, modules
- ✓ Summary of modules
- ✓ Tips for upgrading commands and providers

Remote State

What You'll Learn

- ✓ Why it's useful to read state
- ✓ Locations where state can be stored
- ✓ How to read state from another local project



WARNING

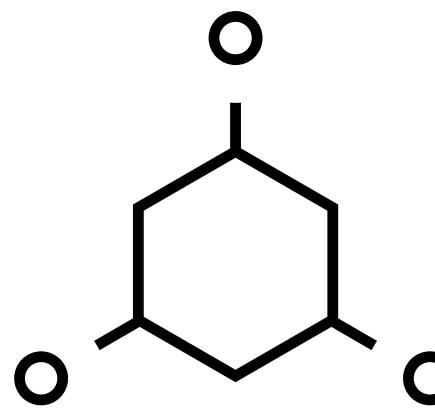
You Know You Need to Refactor Your Monorepo When

An individual can destroy unrelated infrastructure that they don't need to own.

Unneeded resources are defined and created only so that they can be referenced by other resources.

A change to one resource triggers changes to other resources that didn't need to change.

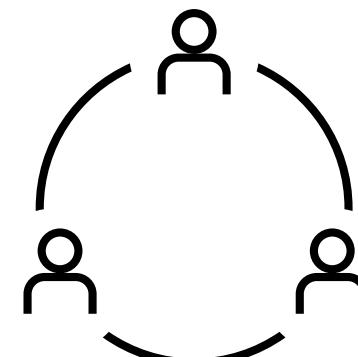
Which of these need to be defined together?



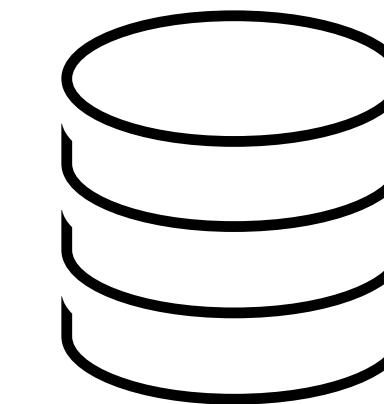
NETWORK



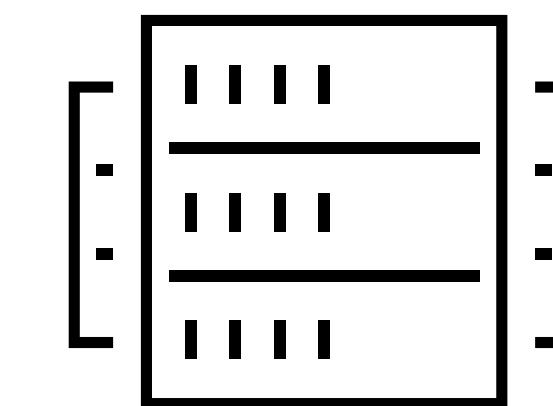
VPC



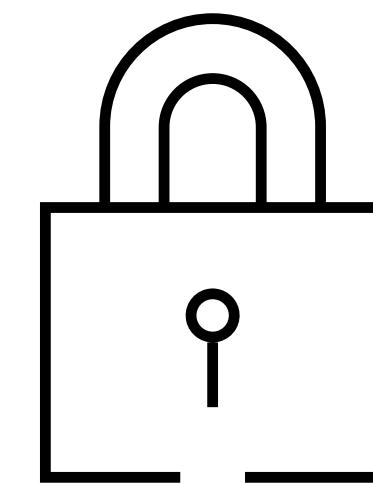
PROFILES



DATABASE

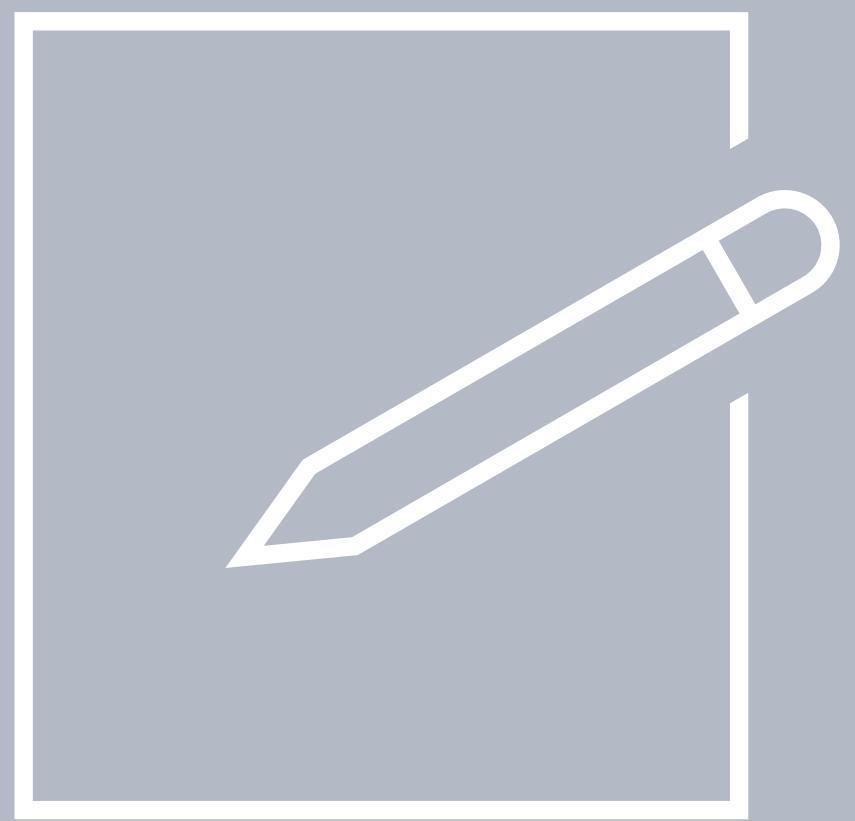


EC2 INSTANCES



KEYS

GOOD



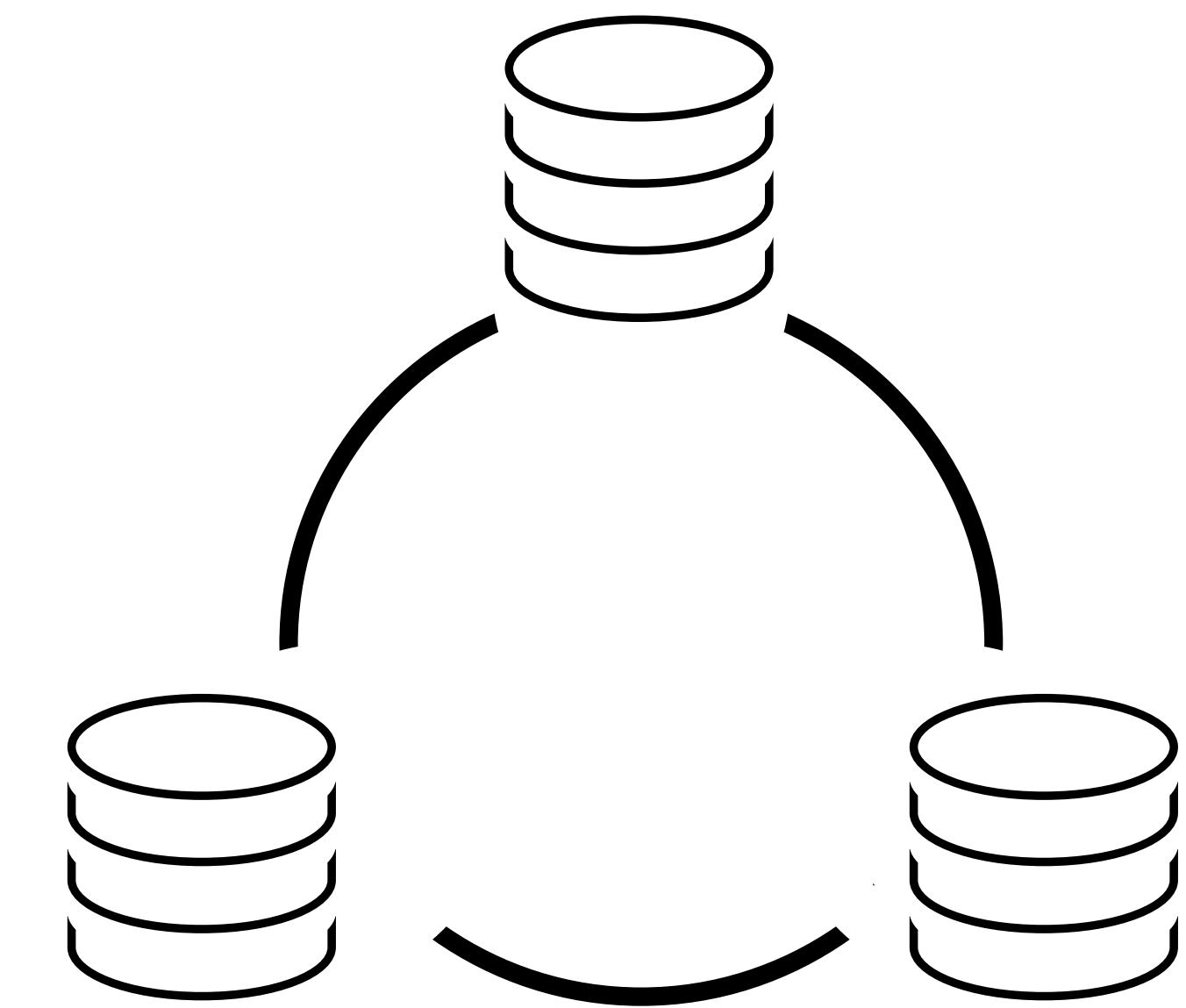
Copy & Paste

BETTER



**Environment
Variables**

BEST



**Shared State
(Outputs)**

```
resource "aws_instance" "workstation" {
    ami              = "${data.aws_ami.app.id}"
    key_name         = "${aws_key_pair.app.id}"

    subnet_id        = "${aws_subnet.app.id}"
    iam_instance_profile = "${aws_iam_instance_profile.app.name}"
    vpc_security_group_ids = ["${aws_security_group.app.id}"]

    user_data = "${data.template_cloudinit_config.app.rendered}"
}
```

*What resources do I need to create
and control?*

*What resources do I need to refer
to when creating my own?*

WARNING

Shortcomings of remote state

Values are read when a configuration is applied, not constantly. A deployment is a snapshot in time.

Remote state can only be read. You cannot write to a remote project or trigger an apply from another one.

Your configuration must have read access to the location where the state file is stored.

State & Its Uses

Collaborate on values from infrastructure created by other projects.

Separating state isolates the impact of Terraform's actions.

Read it locally or on a remote disk (cloud storage) or API (Terraform Enterprise).



Terraform
ENTERPRISE

```
# Default
terraform {
  backend "local" {
    path = "terraform.tfstate"
  }
}
```

```
# Read state from another Terraform config's state
data "terraform_remote_state" "vpc" {
    backend = "local"

    config {
        path = "../shared-vpc/terraform.tfstate"
    }
}
```

```
# Elsewhere, use outputs from the state  
"${data.terraform_remote_state.vpc.cidr_block}"
```

Issues

State files expose sensitive information. Use encryption, use private storage.

Disaster recovery: Use versioning, logging on S3 buckets, if used.

Top level state must explicitly output values for others to read.

*Remote state is read-only.
Remote resources cannot be
created or destroyed.*

EXERCISE

Read State

Create a new Terraform config in a new directory that reads the `public_ip` value from the existing project's state file (from local storage).

Emit it as an output from the new project.

HINT: This requires only about 10 lines of code in total.

What You Learned

- ✓ Why it's useful to read state
- ✓ Where state is stored by default and where it can be stored
- ✓ How to read state from another local project

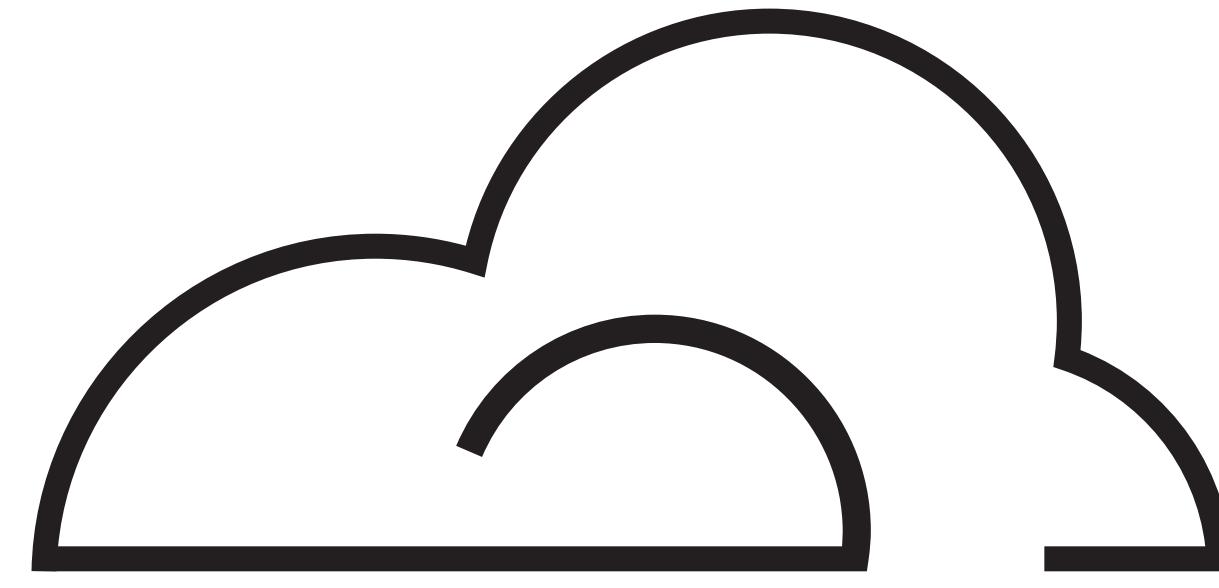
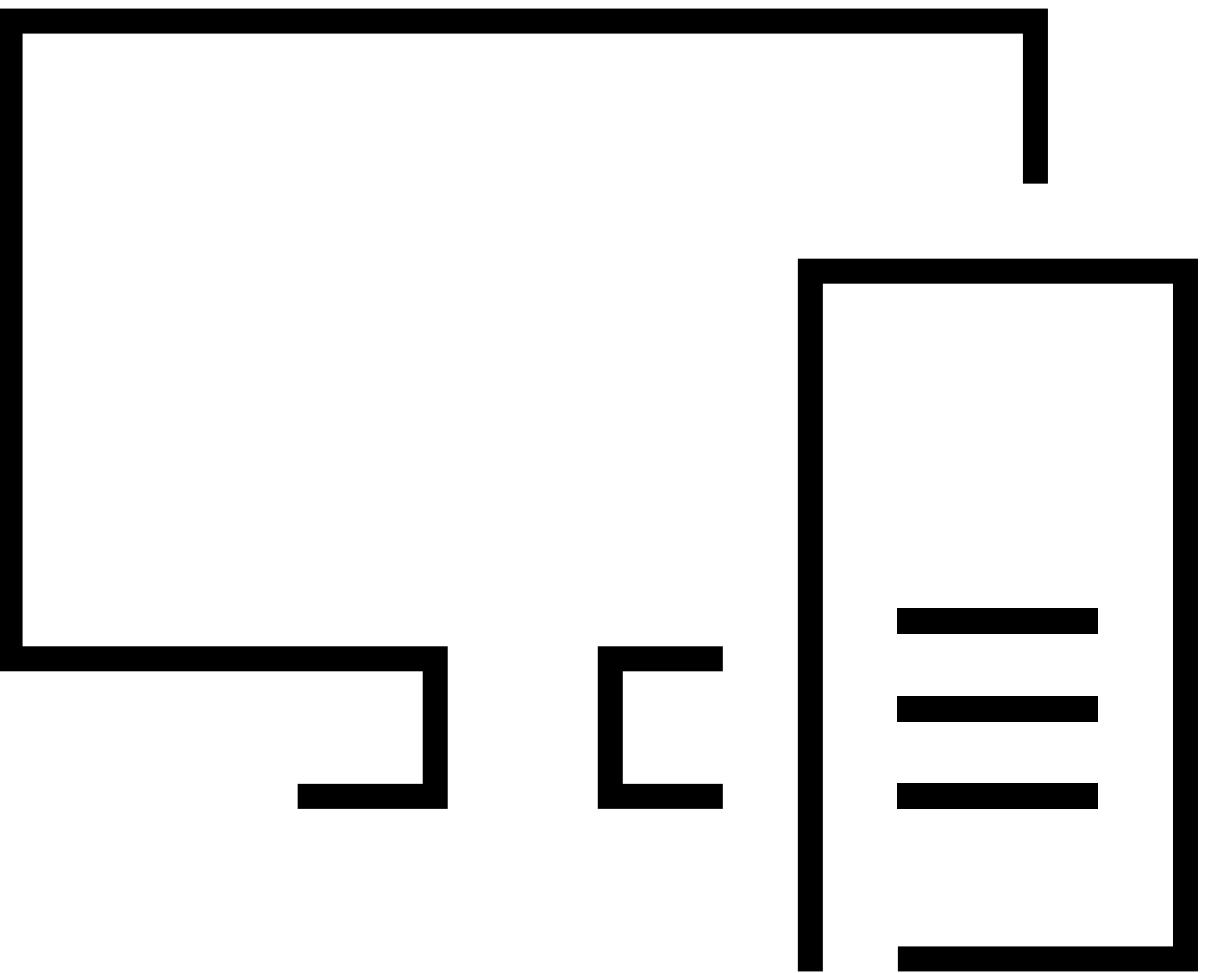
Store State Remotely

What You'll Learn

- ✓ Store state in cloud storage
- ✓ Bootstrap an existing project to store state remotely
- ✓ Initialize a new project to use cloud storage for its state file
- ✓ Read values from remote state

Terraform state is stored in a JSON file. Any config that can access the file can read state from it.

Local vs Remote State



```
export AWS_ACCESS_KEY_ID="abcdef"
export AWS_SECRET_ACCESS_KEY="abcdef"
export AWS_DEFAULT_REGION="us-west-2"
```

```
● ● ●
```

```
terraform {
  backend "s3" {
    bucket  = "engineering-prod-tfstate"
    key     = "root.tfstate"
    encrypt = true
  }
}
```

The screenshot shows the AWS S3 console interface. At the top, the navigation bar includes the AWS logo, 'Services' dropdown, 'Resource Groups' dropdown, user information ('geoffrey @ hashicorp-training'), and 'Global' and 'Support' dropdowns. Below the navigation bar, the breadcrumb path shows 'Amazon S3 > engineering-prod-tfstate'. The main content area features four tabs: 'Overview' (disabled), 'Properties' (selected and highlighted in blue), 'Permissions', and 'Management'. A search bar at the top of the content area contains the placeholder text 'Type a prefix and press Enter to search. Press ESC to clear.' Below the search bar are several action buttons: 'Upload' (blue button), '+ Create folder' (button), 'More' (dropdown menu), 'Versions' (link), 'Hide' (button), and 'Show' (button). To the right of these buttons, the location 'US West (Oregon)' and a refresh icon are displayed. The main list area displays one item:

	Name	Last modified	Size	Storage class
<input type="checkbox"/>	engineering-prod-tfstate-root.tfstate	Apr 27, 2018 8:23:47 PM GMT-0700	4.1 KB	Standard

Type a prefix and press Enter to search. Press ESC to clear.

[Upload](#) [Create folder](#) [More](#) [Versions](#) [Hide](#) [Show](#)

US West (Oregon) [Edit location](#)

Viewing 1 to 2

<input type="checkbox"/>	Name	Last modified	Size	Storage class
<input type="checkbox"/>	engineering-prod-tfstate-root.tfstate	Apr 27, 2018 8:23:47 PM GMT-0700	4.1 KB	Standard
<input type="checkbox"/>	engineering-prod-tfstate-use-state.tfstate	Apr 27, 2018 8:19:41 PM GMT-0700	1.7 KB	Standard

Viewing 1 to 2

Type a prefix and press Enter to search. Press ESC to clear.

[Upload](#) [Create folder](#) [More](#) Versions [Hide](#) [Show](#)

US West (Oregon) [Edit location](#)

Viewing 1 to 8

<input type="checkbox"/>	Name	Version ID	Last modified	Size	Storage class
	engineering-prod-tfstate-root.tfstate		Apr 27, 2018 8:23:47 PM		
<input type="checkbox"/>	Apr 27, 2018 8:23:47 PM (Latest version)	g27jIFSq3JrTetmKMYkmrhP5...		4.1 KB	Standard
<input type="checkbox"/>	Apr 27, 2018 8:20:32 PM	VA8Hx8oPcQC.pqv4aGkd21ru...		4.1 KB	Standard
<input type="checkbox"/>	Apr 27, 2018 8:16:40 PM	rTNee754h1O8lLdjXXU3UpMH...		4.3 KB	Standard
<input type="checkbox"/>	Apr 27, 2018 8:12:00 PM	KDjQJwe2oWVSJlwEV5AQpm...		4.1 KB	Standard
<input type="checkbox"/>	Apr 27, 2018 8:11:48 PM	TsA_lexiwC5cZQCR_D2pgPy5...		4.1 KB	Standard
	engineering-prod-tfstate-use-state.tfstate		Apr 27, 2018 8:19:41 PM		
<input type="checkbox"/>	Apr 27, 2018 8:19:41 PM (Latest version)	3oY6bQ.maPxNjCS842kjbd1kr...		1.7 KB	Standard
<input type="checkbox"/>	Apr 27, 2018 8:16:55 PM	ZpeuWp89Ho2eOFzlhcJBnI6z...		1.7 KB	Standard
<input type="checkbox"/>	Apr 27, 2018 8:15:05 PM	iJXnbx3dxJziCLqqraGMRdvxi...		1.5 KB	Standard

Consider other backends

artifactory

http

azurerm

manta

consul

s3

etcd

swift

etcdv3

terraform enterprise

gcs

```
output "public_ip" {  
    value = "8.8.8.8"  
}  
  
output "destination" {  
    value = {  
        "planet"      = "Mars"  
        "distance"   = "33900000"  
        "date"        = "11 Jan 2058"  
    }  
}
```

*Remote state is read-only.
Resources cannot be created or
destroyed except by the
configuration whose state they
track.*

*Remote shared state does not
need to be merged like files in a
Git repo. State should be locked
and only written by one instance
of Terraform at a time.*

EXERCISE

Work with remote state storage

Create a Terraform config that creates an S3 bucket.
We'll use it to store state.

Store the project's own state on the S3 backend.

Create a second project and read output from the first
project's remote state stored in S3.

Re-emit it as your own output.

What You Learned

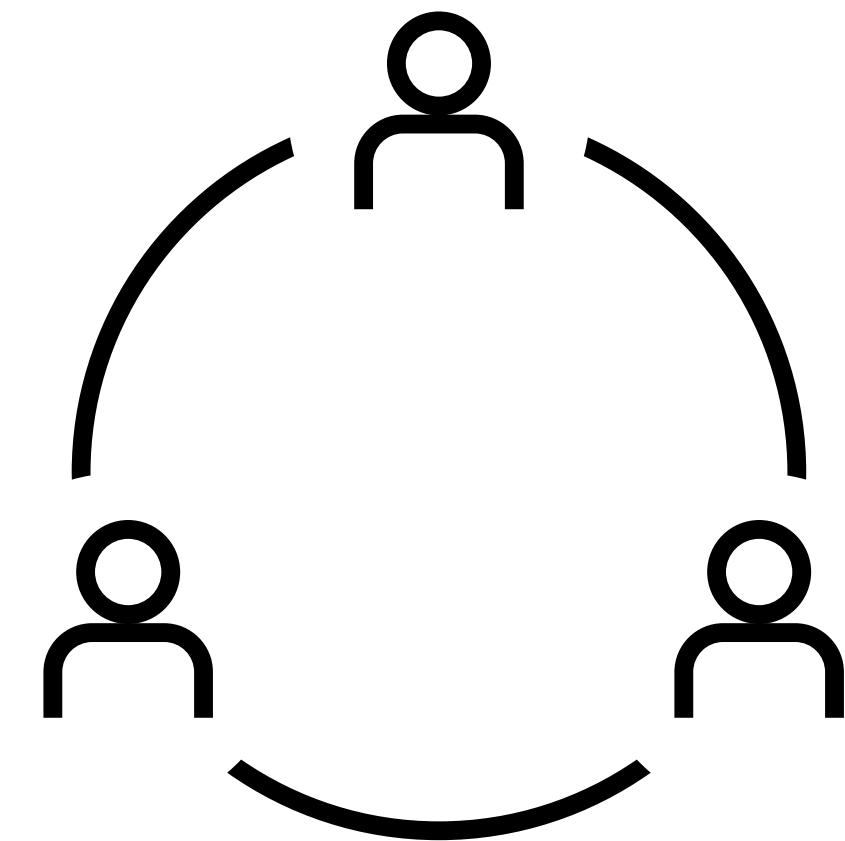
- ✓ Store state in cloud storage
- ✓ Bootstrap an existing project to store state remotely
- ✓ Initialize a new project to use cloud storage for its state file
- ✓ Read values from remote state

Code Organization for Teams

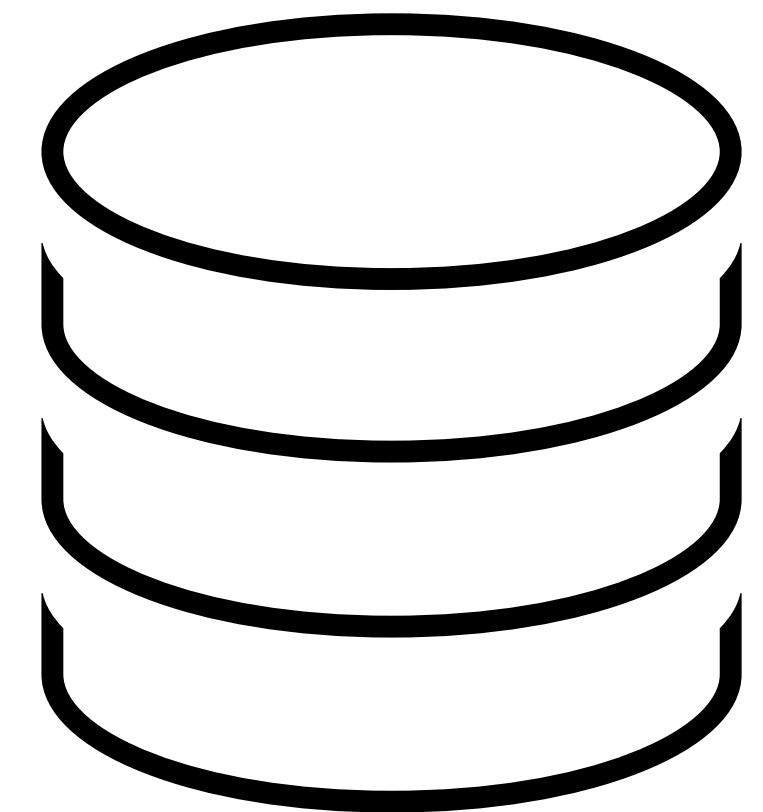
What You'll Learn

- ☑ How Terraform projects evolve
- ☑ When to use modules vs a new project
- ☑ How to use shared state to manage complexity

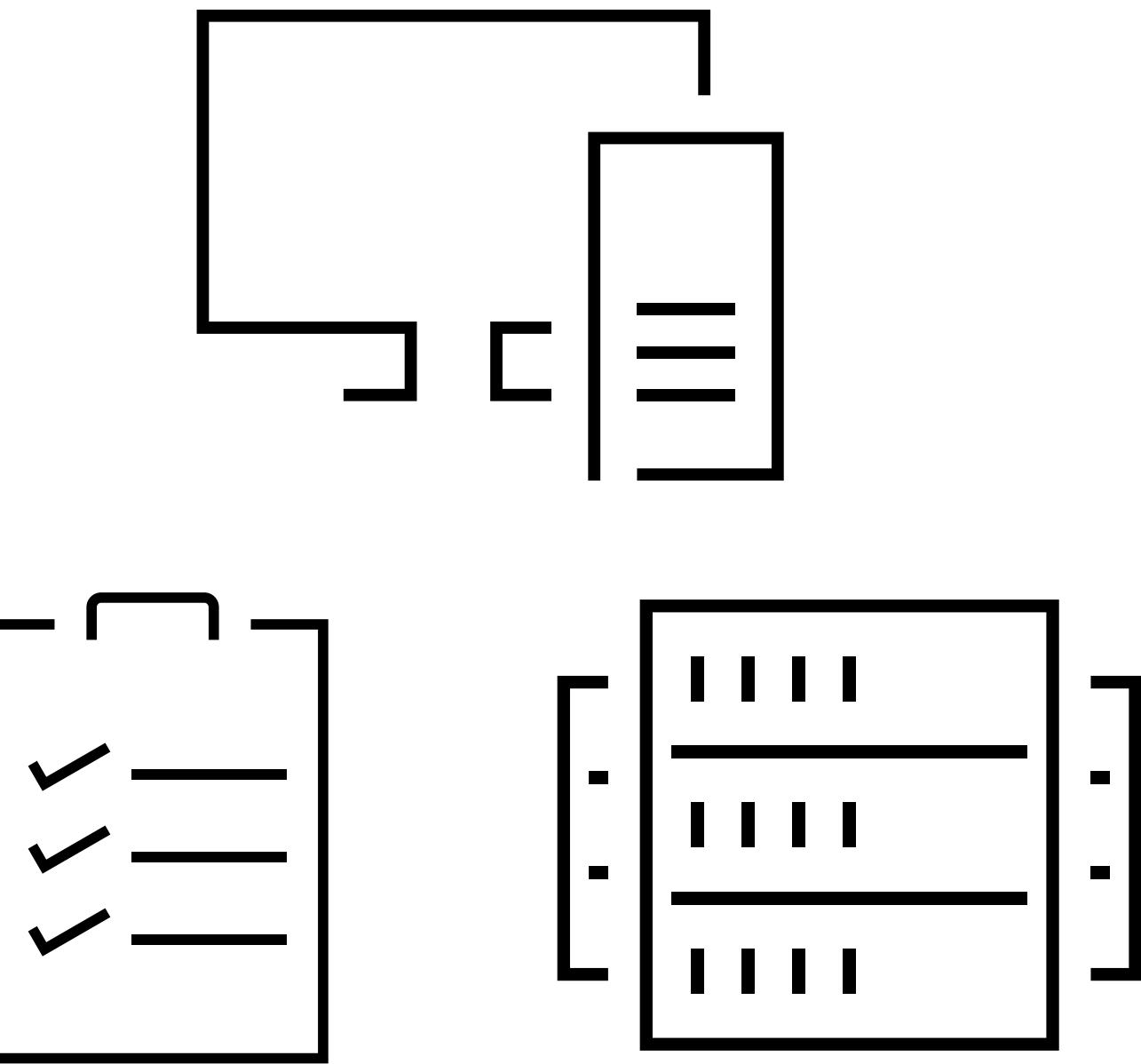
Ways a Project Grows



Larger Team



More Resources



Dev, Test, Staging, Prod



Nicki Watt

Evolving Your Infrastructure with Terraform

[OpenCredo](#)

Terraform Collaboration Questions

Is this config owned by one team?

Is this config used by more than one team?

Does this config create resources which host other resources? (or needed by other resources)

Complexity, Capability

A more expansive infrastructure will need a more complex Terraform project layout, but also requires more communication and additional layers of code management.

Don't start with complexity 7 if you don't need it.



Stable

DRY

Isolated

Reusable

Consistent

Complex

Dynamic

Explicit

Shared

Customizable

Creative

Simple



Many Remote State Files

Single Local State

Versioned Modules

Root Module

Many Repositories

Single Directory

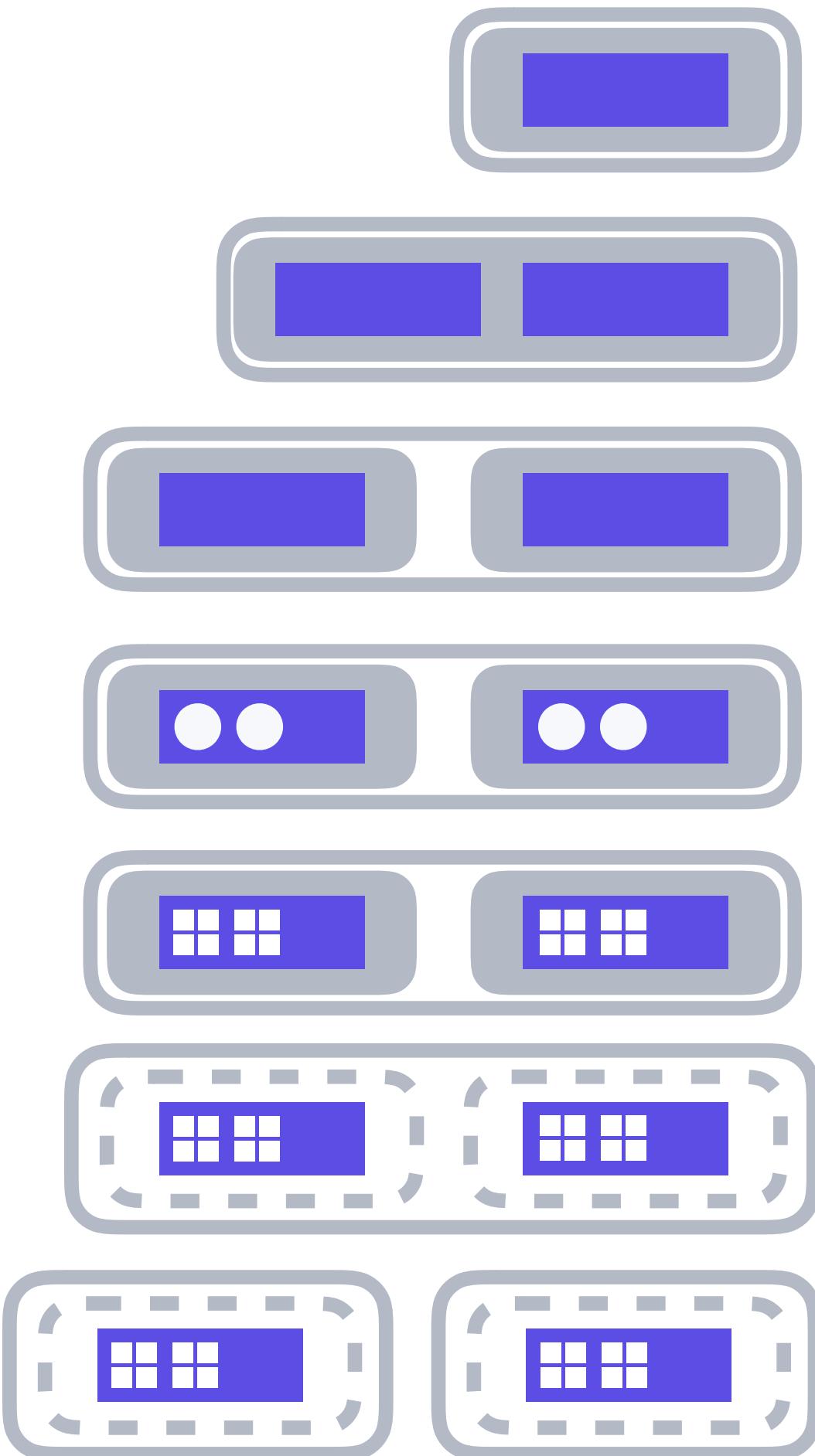
Limited Configuration

Freeform Code

Limited Access

Full Control

Evolution of a Terraform Project



- 1.Single File
- 2.Separate Environments
- 3.State per Environment
- 4.Modules
- 5.Nested Modules
- 6.Distributed State
- 7.Repository Isolation

Single file

main.tf

```
└── 1-single-file
    ├── main.tf
    ├── terraform.tfstate
    └── terraform.tfvars
```

Multiple Environments

May contain test and prod (“web_test”, “web_prod”)

Possibly in different files with a single repo

```
└─ 2-environments
    └─ main-prod.tf
    └─ main-test.tf
    └─ terraform.tfstate
    └─ terraform.tfvars
```

Separate States

Limit the impact of provisioning one configuration

Control timing and release schedule of provisioning

```
└─ 3-states
    └─ prod
        ├ main.tf
        ├ terraform.tfstate
        └─ terraform.tfvars
    └─ test
        ├ main.tf
        ├ terraform.tfstate
        └─ terraform.tfvars
```

Resources in Modules

Improves long term maintenance

Eases reuse

Encourages consistency in design

```
— 4-modules
  |   └── envs
  |       |   └── prod
  |       |       |   └── main.tf
  |       |       |   └── terraform.tfstate
  |       |       └── terraform.tfvars
  |       └── test
  |           |   └── main.tf
  |           |   └── terraform.tfstate
  |           └── terraform.tfvars
  └── modules
      |   └── core
      |       |   └── input.tf
      |       |   └── main.tf
      |       └── output.tf
      └── db
      └── network
      └── webapp
```

Nested Modules

A more complex system outgrows single modules

Able to isolate parts of the system that co-exist but don't depend on each other

```
— 5-nested
  └─ envs
    |   └─ prod
    |   └─ test
    |       └─ core
    |           └─ terraform.tfstate
    |       └─ db
    |           └─ terraform.tfstate
    |       └─ network
    |           └─ terraform.tfstate
    |       └─ webapp
    |           └─ terraform.tfstate
  └─ modules
      └─ common
          └─ aws
              └─ compute
                  |   └─ db
                  |   └─ web
              └─ network
                  └─ priv_subnet
                  └─ pub_subnet
                  └─ vpc
  └─ project
      └─ core
      └─ webapp
```

Distributed State

Necessary for teams

Further isolates individual environments and deployments

```
— 6-distributed
  └─ envs
    |   └─ prod
    |   └─ test
    |       └─ core
    |           └─ terraform.tfstate
    |       └─ db
    |           └─ terraform.tfstate
    |       └─ network
    |           └─ terraform.tfstate
    └─ webapp
        └─ terraform.tfstate
  └─ modules
      └─ common
          └─ aws
              └─ compute
                  └─ db
                  └─ web
              └─ network
                  └─ priv_subnet
                  └─ pub_subnet
                  └─ vpc
  └─ project
      └─ core
      └─ webapp
```

State/Repo Per Component

Full isolation

Control team read/write permissions on each repo

Full control over authorization per state

Limit Terraform Enterprise apply actions

```
— 7-repo-isolation
  └─ envs
    |   └─ prod
    |   └─ test
    |       └─ core
    |           └─ terraform.tfstate
    |       └─ db
    |           └─ terraform.tfstate
    |       └─ network
    |           └─ terraform.tfstate
    └─ webapp
        └─ terraform.tfstate
  └─ modules
      └─ common
          └─ aws
              └─ compute
                  └─ db
                  └─ web
              └─ network
                  └─ priv_subnet
                  └─ pub_subnet
                  └─ vpc
  └─ project
      └─ core
      └─ webapp
```

Terraform Collaboration Answers

Is this config owned by one team?

Make a separate repository. Producing infrastructure

Is this config used by more than one team?

Publish a module. Creating infrastructure

Does this config create resources?

Publish outputs. Existing within infrastructure

Remote State is Read Only

If the cloud provider's API supports adding children to a parent resource, then repository isolation can work as a code organization technique. For example, Google Cloud Platform load balancers expose a tag which can be implemented by child instances in order to join the load balancer.

If a parent resource needs to be modified directly to manage a child resource, then the parent repository may need to be re-deployed (with knowledge of the child's state).

Terraform Enterprise Basics

What You'll Learn

- ✓ Create a repository at GitHub
- ✓ Push your code to GitHub
- ✓ Create an organization at Terraform Enterprise
- ✓ Connect to GitHub via OAuth
- ✓ Add a workspace that references a GitHub repo
- ✓ Commit changes and create infrastructure

Terraform Enterprise Features

Collaboration and Governance

Private install or hosted

SCM integration

Central service registry

Provision at scale

<https://app.terraform.io/account/new>



Create an account

USERNAME

EMAIL

PASSWORD

PASSWORD CONFIRMATION

By clicking on "Create an account" below, you are agreeing to the [Terms of Use](#) and the [Privacy Policy](#).

Create an account

OAuth Application Settings Terraform Enterprise | topfunk | Geoffrey

Secure | https://atlas.hashicorp.com/app/topfunk/settings/oauth

topfunk Workspaces Documentation Status

ORGANIZATION SETTINGS

topfunk ✓

Profile Teams OAuth Configuration API Token User Sessions Manage SSH Keys Sentinel Policy

OAuth Clients

GitHub

Callback URL
<https://atlas.hashicorp.com/auth/94dc3315-5f10-42af-8e04-7d95b17dcd8a/callback>

HTTP URL
<https://github.com>

API URL
<https://api.github.com>

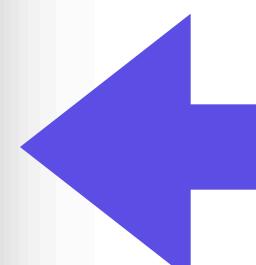
Created
Dec 12, 2017 17:22:54PM

A connection was made on Dec 12, 2017 17:26:01PM by authenticating via OAuth as GitHub user **topfunk**, which assigned an OAuth token for use by all Terraform Enterprise users in the **topfunk** organization.

You can [add a private SSH key](#) to this connection to be used for git clone operations.

[Revoke Connection](#)

[Delete Client](#)



OAuth Application Settings Terraform Enterprise | topfunk

Secure | https://atlas.hashicorp.com/app/topfunky/workspaces/new

Geoffrey

topfunky Workspaces Documentation | Status

Create a new Workspace

This workspace will be created under the current organization, **topfunky**.

WORKSPACE NAME

training

The name of your workspace is unique and used in tools, routing, and UI. Dashes and alphanumeric characters are permitted. Learn more about [naming workspaces](#).

VCS CONNECTION

GitHub

Don't see the VCS connection you're looking for? [Configure a different VCS connection](#).

REPOSITORY

|

topfunky/training

OAuth Application Settings Terraform Enterprise | topfunk... Geoffrey

Secure | https://atlas.hashicorp.com/app/topfunky/training/variables

topfunky Workspaces Documentation Status

training Queue Plan

Latest Run Runs States Variables Settings Integrations Access

Terraform Variables

These variables are set using the `var` option when performing a plan and apply

Editing

access_key	xxxxxxxxxx	<input type="checkbox"/> HCL <input checked="" type="checkbox"/> Sensitive	
secret_key	xxxxxxxxxx	<input type="checkbox"/> HCL <input checked="" type="checkbox"/> Sensitive	
ami	xxxxx	<input type="checkbox"/> HCL <input type="checkbox"/> Sensitive	
subnet_id	xxxxx	<input type="checkbox"/> HCL <input type="checkbox"/> Sensitive	
vpc_security_group_id	xxxxx	<input type="checkbox"/> HCL <input type="checkbox"/> Sensitive	
identity	xxxxx	<input type="checkbox"/> HCL <input type="checkbox"/> Sensitive	

Terraform Enterprise | topfunk... | Use variables for public and pr | Geoffrey

GitHub, Inc. [US] | https://github.com/topfunkym-demo/terraform-intro-demo/pull/2

No description provided.

topfunkym-demo added some commits 4 minutes ago

- Use variables for public and private key Verified e4fd78a
- Expose variables for server module Verified 8b27105

Add more commits by pushing to the **topfunkym-demo-patch-2** branch on **topfunkym-demo/terraform-intro-demo**.

All checks have failed Hide all checks

1 failing check

atlas/topfunkym-demo/terraform-intro-demo — Terraform plan errored Details

This branch has no conflicts with the base branch
Merging can be performed automatically.

Merge pull request ▼ You can also [open this in GitHub Desktop](#) or view [command line instructions](#).

Write Preview AA B i “ <> ↵ I E ⌂ ↵ @ ★

Leave a comment

Attach files by dragging & dropping, [selecting them](#), or pasting from the clipboard.

Assignees: No one—assign yourself

Labels: None yet

Projects: None yet

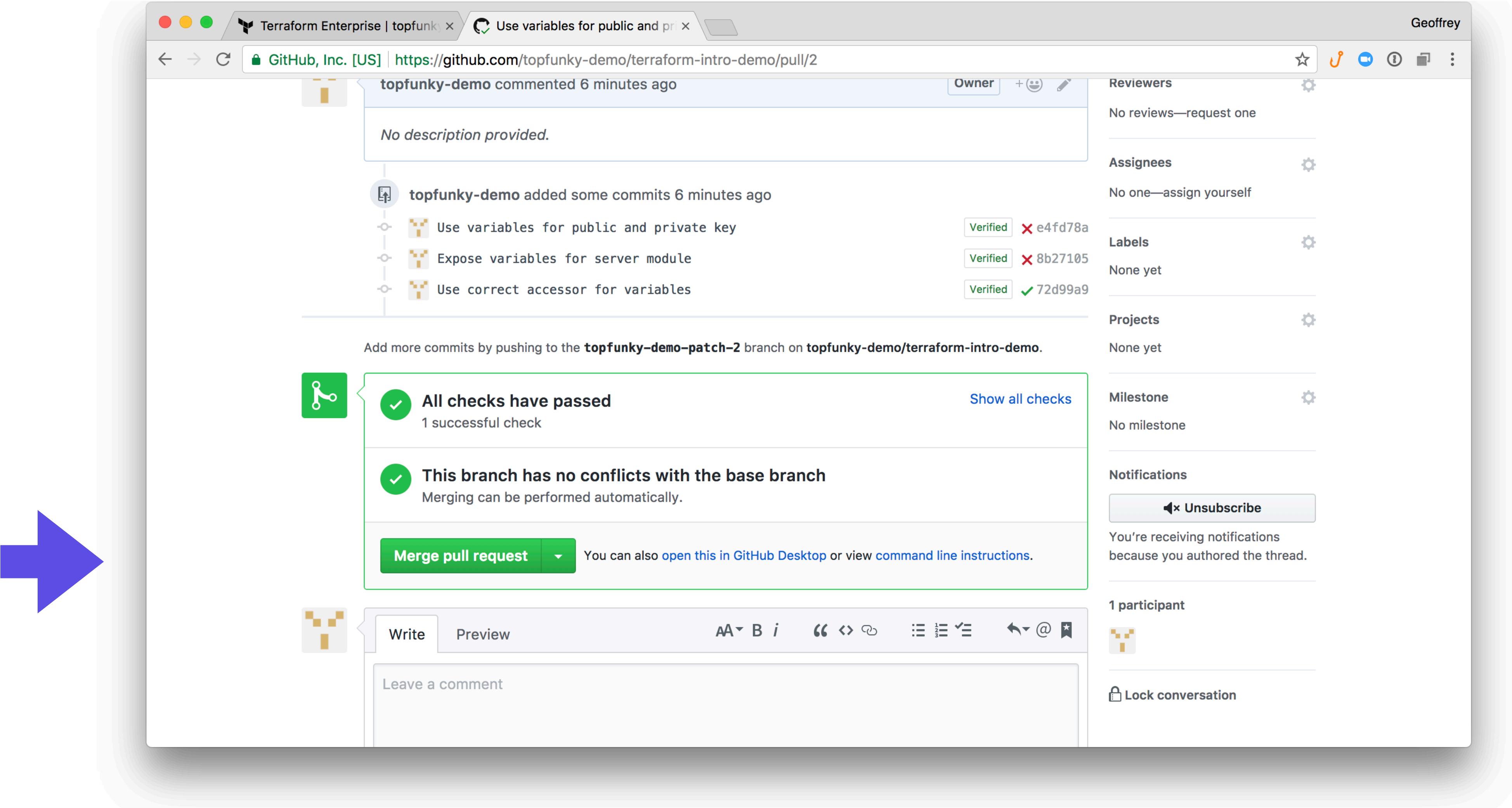
Milestone: No milestone

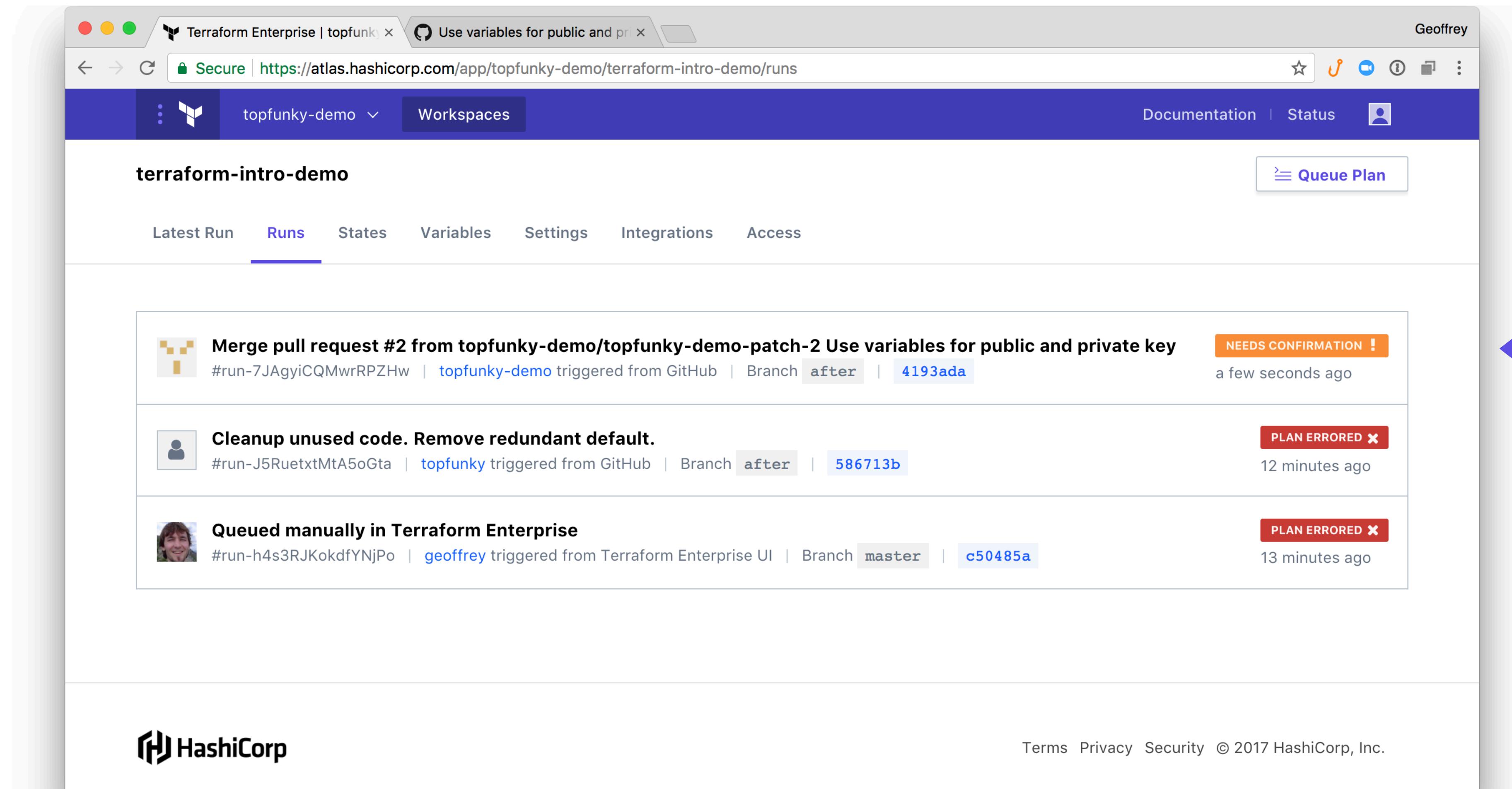
Notifications: [Unsubscribe](#)
You're receiving notifications because you authored the thread.

1 participant:

Lock conversation

The screenshot shows a GitHub pull request page for a Terraform repository. The pull request has no description. It contains two commits from the user 'topfunkym-demo'. The first commit, 'Use variables for public and private key', is verified and has a commit hash of 'e4fd78a'. The second commit, 'Expose variables for server module', is also verified and has a commit hash of '8b27105'. A note at the bottom of the commit list says to add more commits by pushing to the 'topfunkym-demo-patch-2' branch. Below the commits, there is a prominent yellow box containing a 'All checks have failed' message, indicating a Terraform plan error. The error message is 'atlas/topfunkym-demo/terraform-intro-demo — Terraform plan errored'. A note below the error says 'This branch has no conflicts with the base branch' and suggests automatic merging. At the bottom of the yellow box are buttons for 'Merge pull request' and links to open it in GitHub Desktop or view command line instructions. The right side of the screen shows standard GitHub pull request sidebar settings for assignees, labels, projects, milestones, notifications, and participants. The participant listed is 'Geoffrey'.





Terraform Enterprise | topfunk... | Use variables for public and pr... Geoffrey

Secure | https://atlas.hashicorp.com/app/topfunky-demo/terraform-intro-demo/runs

topfunky-demo Workspaces Documentation | Status

terraform-intro-demo Queue Plan

Latest Run Runs States Variables Settings Integrations Access

Merge pull request #2 from topfunky-demo/topfunky-demo-patch-2 Use variables for public and private key NEEDS CONFIRMATION !
#run-7JAgyiCQMwrRPZHW | topfunky-demo triggered from GitHub | Branch after | 4193ada a few seconds ago

Cleanup unused code. Remove redundant default. PLAN ERRORED ✘
#run-J5RuetxtMtA5oGta | topfunky triggered from GitHub | Branch after | 586713b 12 minutes ago

Queued manually in Terraform Enterprise PLAN ERRORED ✘
#run-h4s3RJKokdfYNjPo | geoffrey triggered from Terraform Enterprise UI | Branch master | c50485a 13 minutes ago

HashiCorp Terms Privacy Security © 2017 HashiCorp, Inc.

Terraform Enterprise | topfunk... x Use variables for public and pr x Geoffrey

Secure | https://atlas.hashicorp.com/app/topfunky-demo/terraform-intro-demo/runs/run-7JAgyiCQMwrRPZHw

Variables input to this run
Configuration input to this run

Run created by GitHub Webhook

PLAN

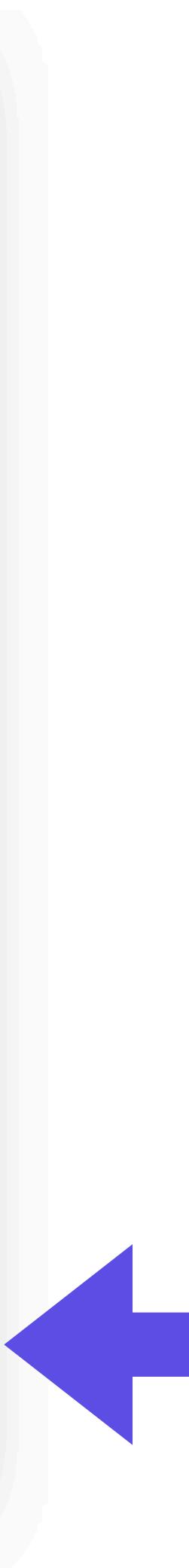
Queued a minute ago
Started a minute ago

Executed and saved successfully a minute ago [View Plan](#)

State versions output with this plan
No state versions output

Leave a comment

[Confirm & Apply](#) [Discard Plan](#)



Terraform Enterprise | topfunk... | Use variables for public and pr... Geoffrey

Secure | https://atlas.hashicorp.com/app/topfunky-demo/terraform-intro-demo/runs/run-7JAgyiCQMwrRPZHw

APPLY

Queued a few seconds ago
Started a few seconds ago

Executed successfully with changes
a few seconds ago

Hide Apply

[View raw log](#)

```
module.server.aws_instance.web (remote-exec):  Private key: true
module.server.aws_instance.web (remote-exec):  SSH Agent: false
module.server.aws_instance.web (remote-exec): Connected!
module.server.aws_instance.web: Creation complete after 27s (ID: i-0936d03b68874f0b6)

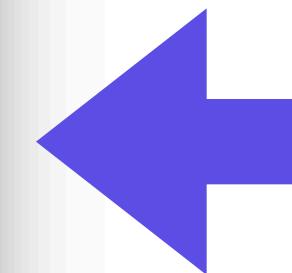
Apply complete! Resources: 2 added, 0 changed, 0 destroyed.

Outputs:

public_dns = [
    ec2-34-212-49-75.us-west-2.compute.amazonaws.com
]
public_ip = [
    34.212.49.75
]
```

State versions output with this apply

- topfunky-demo/terraform-intro-demo#sv-G8QTDAv9vuHKzDqA
- topfunky-demo/terraform-intro-demo#sv-LdHR89hU5K9sEmNM



EXERCISE

Connect GitHub to TFE and Fork a GitHub Repo

Fork the GitHub repository at

`hashicorp/demo-terraform-101`

Connect Terraform Enterprise to GitHub

Create variables on Terraform Enterprise

What You Learned

- ✓ Create a repository at GitHub
- ✓ Push your code to GitHub
- ✓ Create an organization at Terraform Enterprise
- ✓ Connect to GitHub via OAuth
- ✓ Add a workspace that references a GitHub repo
- ✓ Commit changes and create infrastructure

Collaborating with TFE

What You'll Learn

- ☑ Understand the role of Terraform Enterprise in modern continuous provisioning workflows
- ☑ Understand the components of Terraform Enterprise: users, organizations, teams, workspaces

TIP

Follow Along in Your Terraform Enterprise Account

Initial setup of a Terraform Enterprise account is quick and easy.

Full import of repositories, code, users, and teams can take longer.

Follow along with this chapter as much as you can with the resources that have been setup in your specific account.

NOTE: Private installations of Terraform Enterprise are nearly identical to what you'll see here.

Terraform Enterprise | hashicorp X +

hashicorp-training Workspaces Modules Documentation | Status

Workspaces 4 total + New Workspace

All (4) Success (1) Error (0) Needs Attention (0) Running (0) Search by name

WORKSPACE NAME	RUN STATUS	LATEST CHANGE	RUN	REPO
training-partner-team-prod	✓ APPLIED	17 days ago	run-9msm	hashicorp/training-partner-team
training-lab-geoffrey				hashicorp/training

What Is Terraform Enterprise?

Platform for collaboration and governance with Terraform

Central module registry

Connection to version control for Terraform configuration code

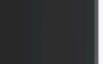
Runs Terraform plan and apply

Log of provisioning actions

Stores state securely

Shares state within organizations

Configurable permissions for user and team access



Open Source



Provision any infrastructure

[Download Free](#)

Infrastructure as code [?](#)

Separation of infrastructure plans and applies [?](#)

Multi-provider support [?](#)

Public module registry [?](#)

Pro ENTERPRISE



Collaboration and operations features for teams

[Request Info](#)

All Open Source features plus

Workspace management and GUI [?](#)

Version control connection [?](#)

Secure variable management [?](#)

Remote plans, applies, state storage, locking, state rollback [?](#)

Premium ENTERPRISE



Governance and policy features for organizations

[Request Info](#)

All Pro features plus

Policy as code management [?](#)

SSO with SAML [?](#)

Track infrastructure changes with audit logs [?](#)

Private install [?](#)

Terraform Enterprise | hashicorp X +

hashicorp-training Workspaces Modules Documentation | Status

Workspaces 4 total + New Workspace

All (4) Success (1) Error (0) Needs Attention (0) Running (0) Search by name

WORKSPACE NAME	RUN STATUS	LATEST CHANGE	RUN	REPO
training-partner-team-prod	✓ APPLIED	17 days ago	run-9msm	hashicorp/training-partner-team
training-lab-geoffrey				hashicorp/training

Terraform Enterprise | geoffrey X +

geoffrey-org Workspaces Modules Documentation Status

training-lab-dev

Queue Plan

Current Run Runs States Variables Settings Version Control Access

run-3U6i triggered from Terraform Enterprise UI 4 days ago

✓ APPLIED 4 days ago

Use updated repo for code

By  geoffrey-hashicorp

Commit [d756a0b](#)

Branch master

Repo [geoffrey-hashicorp/training-lab](#)

Run Timeline



Variables input to this run



Configuration input to this run



Run created by  geoffrey

Queued from Terraform Enterprise UI



APPLY

Queued 4 days ago

Started 4 days ago

Executed successfully with changes

4 days ago

[Hide Apply](#)

[View raw log](#)

[Top](#)

[Bottom](#)

[+](#)

[↗](#)

```
servers = [
  35.166.219.112,
  54.200.186.220,
  54.244.179.158
]
vault-lb = demo-geoffrey-vault-40338213.us-west-2.elb.amazonaws.com
vault-root-token = demo-geoffrey-87527beded15210f
workstations = [
  18.236.126.204,
  34.217.101.207,
  54.187.133.88,
  35.164.123.169,
  34.214.7.120
]
```

State versions output with this apply

- [geoffrey-org/training-lab-dev#sv-gS4yfAF7v6NQJzyT](#)
- [geoffrey-org/training-lab-dev#sv-d1K1AnjNCfgMQy1j](#)

Terraform Enterprise | topfunky | Use variables for public and pri | Geoffrey

GitHub, Inc. [US] | https://github.com/topfunky-demo/terraform-intro-demo/pull/2

topfunky-demo commented 6 minutes ago

No description provided.

topfunky-demo added some commits 6 minutes ago

- Use variables for public and private key Verified ✘ e4fd78a
- Expose variables for server module Verified ✘ 8b27105
- Use correct accessor for variables Verified ✓ 72d99a9

Add more commits by pushing to the **topfunky-demo-patch-2** branch on **topfunky-demo/terraform-intro-demo**.

All checks have passed 1 successful check

This branch has no conflicts with the base branch Merging can be performed automatically.

Merge pull request ▾ You can also open this in GitHub Desktop or view command line instructions.

Write Preview AA B i “ < > @ Lock conversation

Leave a comment

Reviewers
No reviews—request one

Assignees
No one—assign yourself

Labels
None yet

Projects
None yet

Milestone
No milestone

Notifications
Unsubscribe
You're receiving notifications because you authored the thread.

1 participant

Lock conversation

User

Individual account on app.terraform.io

Can be a member of multiple teams and organizations

Signup is global at:

<https://app.terraform.io/account/new>

Usernames are global across all [terraform.io](https://app.terraform.io) users



Create an account

USERNAME

EMAIL

PASSWORD

PASSWORD CONFIRMATION

By clicking on "Create an account" below, you
are agreeing to the [Terms of Use](#) and the
[Privacy Policy](#).

[Create an account](#)

Organization

Shared spaces for teams to collaborate on workspaces.

Organizations can have many teams.

Sentinel policies run at the organization level across all workspaces.

Terraform Enterprise | hashicorp X +

hashicorp-training Workspaces Modules Documentation | Status

Workspaces hashicorp-training ✓ + New Workspace

All (4) Organization Settings

WORKS SWITCH ORGANIZATIONS

geoffrey-org topfunky-demo

Create new organization

Organization hashicorp-training ✓

Needs Attention (0) Running (0)

Search by name

RUN STATUS	LATEST CHANGE	RUN	REPO
✓ APPLIED	17 days ago	run-9msm	hashicorp/training-partner-team
			hashicorp/training

HashiCorp

Team

Groups of users that reflect your company's organization or project structure.

Teams can be granted read, write, and admin permissions to workspaces.

Teams are created under the organization settings.

Permissions are configured under workspace access settings.



ORGANIZATION SETTINGS

hashicorp-training ✓

Profile

Teams

VCS Provider

API Token

Authentication

Manage SSH Keys

Sentinel Policy

Team Management

Teams let you group users into specific categories to allow for finer grained access control policies. For example, your developers could be on a dev team that only has access to applications.

In order to allow a team access to a resource, go to the Access settings for the specific resource and enter the team name. At this point you can control the access level for that team.

The **owners** team is a special team that has implied access for all of your resources, but also has the ability to manage your organization.

Create new team

NAME

 Team nameCreate team

The name of the team.

Teams

owners	6 members
--------	-----------

Workspaces

Workspaces are how Terraform Enterprise tracks infrastructure.

One Terraform configuration linked to a version control repository.

Values for variables used by the configuration.

Persistent stored state.

Historical state and run logs.

Workspace Approaches

One approach is to have one workspace per environment per config, named accordingly.

- billing-app-dev
- billing-app-stage
- billing-app-prod
- networking-dev
- networking-stage
- networking-prod

Alternate Workspace Approaches

Or, you could create a separate organization for development, test, or staging environments.

Organizations cannot share state, so this keeps them isolated.

Variables

Terraform Enterprise variables replace both `terraform.tfvars` and any environment variables that you would use on a local machine.

Variables can be marked as *sensitive*, in which case they will not be displayed in the UI on subsequent visits.

Variables must be saved before they will be used.

training-lab-dev

Queue Plan

Current Run Runs States Variables Settings Version Control Access

Terraform Variables

These variables are set using the `var` option when performing a plan and apply

Editing

access_key	sensitive - write only	<input type="checkbox"/> HCL <input checked="" type="checkbox"/> Sensitive	
namespace	demo-geoffrey	<input type="checkbox"/> HCL <input type="checkbox"/> Sensitive	
public_key	sensitive - write only	<input type="checkbox"/> HCL <input checked="" type="checkbox"/> Sensitive	
region	us-west-2	<input type="checkbox"/> HCL <input type="checkbox"/> Sensitive	
secret_key	sensitive - write only	<input type="checkbox"/> HCL <input checked="" type="checkbox"/> Sensitive	
training_password	rumplestiltskin	<input type="checkbox"/> HCL <input type="checkbox"/> Sensitive	
training_username	grey_wolf	<input type="checkbox"/> HCL <input type="checkbox"/> Sensitive	
workstations	5	<input type="checkbox"/> HCL <input type="checkbox"/> Sensitive	

How your team works

TFE can be used as a part of your team's existing workflows, or you can evolve your team workflows to take advantage of TFE features.

- **Manual:** Queue runs on demand
- **Semi-auto:** plan runs on VCS commit
- **Continuous Infrastructure:** apply on VCS commit

API

Terraform Enterprise provides a comprehensive API. Many companies drive workflows entirely through the API.

Due to the uniqueness of applications and the depth of features offered, we won't discuss it here.

See the documentation for more details:

<https://www.terraform.io/docs/enterprise/api/index.html>

API Tokens

Three kinds of API tokens are offered: organization, team, or user.

Accessing state from outside TFE requires a user token.

Team tokens are useful for triggering plans and applies from external CI/CD tools.

Organization tokens only modify teams and workspaces. They cannot trigger plan or apply.

```
data "terraform_remote_state" "vpc" {  
    backend = "atlas"  
    config {  
        name = "phoenixbank/vpc-prod"  
    }  
}  
  
# Elsewhere, use outputs from the state  
"${data.terraform_remote_state.vpc.cidr_block}"
```

What You Learned

- ☑ Understand the role of Terraform Enterprise in modern continuous provisioning workflows
- ☑ Understand the components of Terraform Enterprise: users, organizations, teams, workspaces

Template File Provider

What You'll Learn

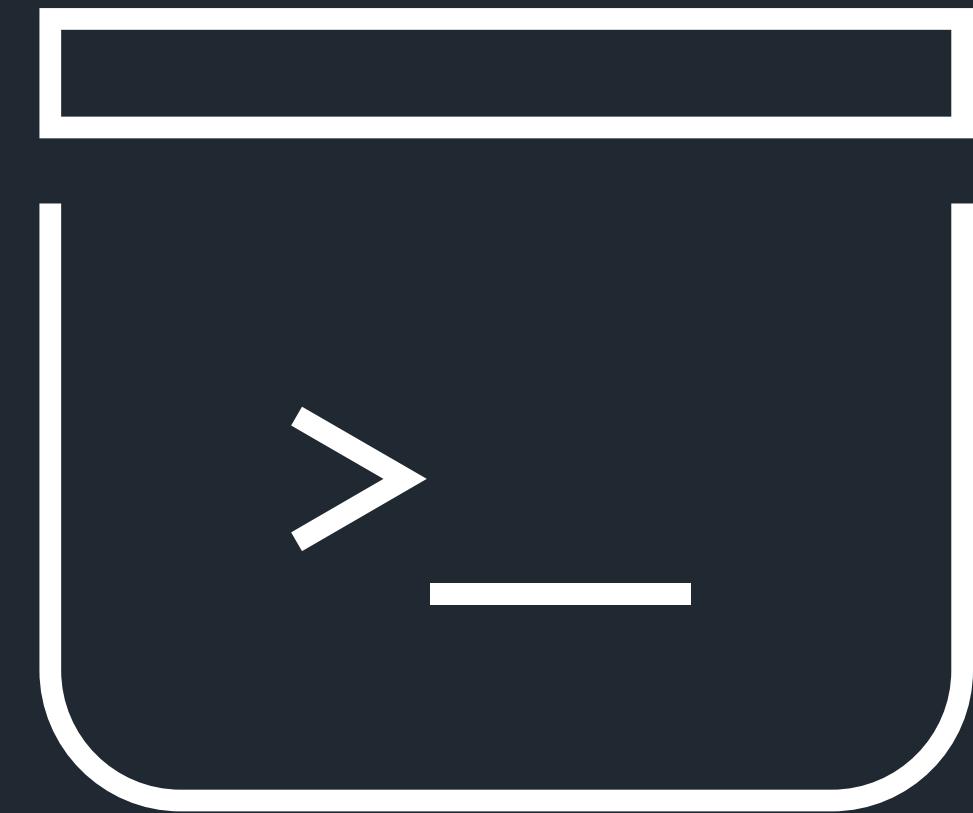
- ✓ Use the template_file provider
- ✓ Render an IAM policy from a template
- ✓ Understand real world scenarios for using templates



Plain Text
Template



Variables from Cloud or
Other Resources



Render with
Terraform

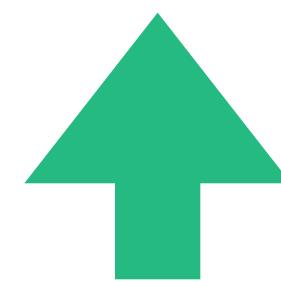
```
~
```

```
$ mkdir -p tfproject/templates && cd $_
```

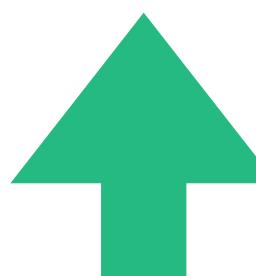
```
~/tfproject/templates
```

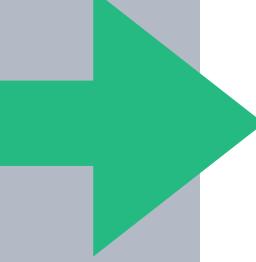
```
$ touch iampolicy.tpl.json
```

```
{  
  "Version": "2012-10-17",  
  "Statement": [  
    {  
      "Sid": "RestrictInstanceLifecycle",  
      "Effect": "Allow",  
      "Action": [  
        "ec2:StartInstances",  
        "ec2:StopInstances",  
        "ec2:TerminateInstances"  
      ],  
      "Resource": [  
        "arn:aws:ec2:us-west-1a:1234567890:instance/*"  
      ]  
    }  
  ]  
}
```



```
{  
  "Version": "2012-10-17",  
  "Statement": [  
    {  
      "Sid": "RestrictInstanceLifecycle",  
      "Effect": "Allow",  
      "Action": [  
        "ec2:StartInstances",  
        "ec2:StopInstances",  
        "ec2:TerminateInstances"  
      ],  
      "Resource": [  
        "arn:aws:ec2:${region}:${owner_id}:instance/*"  
      ]  
    }  
  ]  
}
```





```
variable "region" {
  default = "us-west-2"
}

variable "owner_id" {
  default = "1234567890"
}

data "template_file" "iam_policy" {
  template = "${file("templates/iam_policy.tpl.json")}"

  vars {
    region      = "${var.region}"
    owner_id   = "${var.owner_id}"
  }
}

output "iam_policy" {
  value = "${data.template_file.iam_policy.rendered}"
}
```

```
~/tfproject
$ terraform init
Terraform has been successfully initialized!
```

```
$ terraform apply  
data.template_file.iam_policy: Refreshing state...
```

```
Apply complete! Resources: 0 added, 0 changed, 0 destroyed.
```

Outputs:

```
iam_policy = {  
    "Version": "2012-10-17",  
    "Statement": [  
        {  
            "Sid": "RestrictInstanceLifecycle",  
            "Effect": "Allow",  
            "Action": [  
                "ec2:StartInstances",  
                "ec2:StopInstances",  
                "ec2:TerminateInstances"  
            ],  
            "Resource": [  
                "arn:aws:ec2:us-west-2:123456789:instance/*"  
            ]  
        }  
    ]  
}
```

Real world scenarios

Pass custom scripts to an instance's `user_data`

Create a new IAM policy with the `policy` argument

Customize a configuration file, such as a startup script or a Consul config

```
resource "aws_iam_policy" "policy" {
    name = "service_policy"
    path = "/"
    description = "Service policy"

    policy = "${data.template_file.policy.rendered}"
}
```

```
resource "aws_instance" "web" {  
    ami = "ubuntu"  
    instance_type = "t2.micro"  
  
    user_data = "${data.template_file.user_data.rendered}"  
}
```

```
data "template_file" "init" {
    template = "${file("${path.module}/init.tpl")}"

    vars {
        consul_address = "${aws_instance.consul.private_ip}"
    }
}
```

TIP

Build Absolute Paths With `path.module`

When rendering templates from within modules, it's possible to end up in the wrong directory.

To fix this, use `path.module` when supplying the path to a template. This builds an absolute path to the current module.

```
template = "${file("${path.module}/init.tpl")}"
```

EXERCISE

Write and Render a Template

Create a template file to reference an IAM Policy

Edit variables of template for Terraform to interpolate

Pass template files into Terraform configurations with data providers

Visualize actual output after applying new policy

Packer for Machine Images

What You'll Learn

- ☑ The workflow for building images with Packer
- ☑ How to use Packer JSON with cloud image APIs
- ☑ How to copy files and run scripts
- ☑ Tips and tricks for achieving a successful result quickly and without frustration

WARNING

It's Too Easy To Rely Too Much on Terraform Provisioners

Provisioners were designed to defer to a configuration management system, not run 700 lines of user data shell scripts.

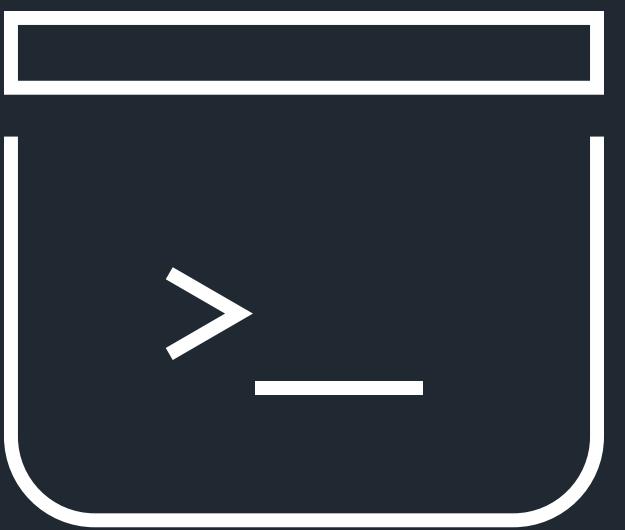
Terraform was not designed to report errors that occur inside machine images (`cloud_init.log`).

Modifying existing images line by line at provision time is slow.

The solution is to build custom images with Packer.



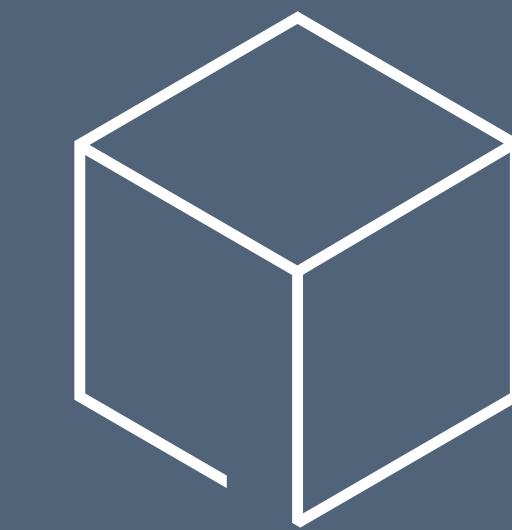
JSON



Build



EC2



ami-d3f8ae

EC2 Management Console X +

https://us-west-2.console.aws.amazon.com/ec2/v2/home?region=us-west-2#Images:sort=Name

aws Services Resource Groups 🔍 geoffrey @ hashicorp-training Oregon Support

EC2 Dashboard Events Tags Reports Limits

INSTANCES Instances Launch Templates Spot Requests Reserved Instances Dedicated Hosts Scheduled Instances

IMAGES AMIs **Bundle Tasks**

ELASTIC BLOCK STORE Volumes Snapshots

NETWORK & SECURITY Security Groups

Launch Actions

Owned by me Filter by tags and attributes or search by keyword

AMI Name AMI ID Source Owner Visibility Status Created

AMI Name	AMI ID	Source	Owner	Visibility	Status	Created
geoffrey-tmp-1527132492	ami-94334aec	130490850807/...	130490850807	Private	available	May
packer-demo-1516924133	ami-e70db29f	130490850807/...	130490850807	Private	available	Jan
packer-demo-1523410128	ami-2dabcb55	130490850807/...	130490850807	Private	available	April

Image: ami-94334aec

Details Permissions Tags

Add/Edit Tags

Key	Value	Show Column
Created-by	Packer	Show Column
OS_Version	Ubuntu	Show Column
Release	Latest	Show Column

```
$ packer build web.json

==> amazon-ebs: Provisioning with shell script: /var/
folders/qt/yn95gld92fjf881q3b018c900000gn/T/packer-
shell120046678
    amazon-ebs: fatal: destination path '/home/ubuntu'
already exists and is not an empty directory.
==> amazon-ebs: Terminating the source AWS instance...
==> amazon-ebs: Cleaning up any extra volumes...
==> amazon-ebs: No volumes to clean up, skipping
==> amazon-ebs: Deleting temporary security group...
==> amazon-ebs: Deleting temporary keypair...
Build 'amazon-ebs' errored: Script exited with non-zero exit
status: 128

==> Some builds didn't complete successfully and had errors:
--> amazon-ebs: Script exited with non-zero exit status: 128

==> Builds finished but no artifacts were created.
```

Packer Syntax & Features

Declared as JSON.

Variables can be provided on the command line or in the file. Predates Terraform so the syntax is slightly different.

Packer can build images for multiple platforms (AWS, Azure, GCP).

Provisioners handoff to a configuration management tool or run shell commands or scripts.

Builders

Alicloud ECS

File

Null

QEMU

Amazon EC2

Google Cloud

1&1

Scaleway

Azure

Hyper-V

OpenStack

Triton

CloudStack

LXC

Oracle

VirtualBox

DigitalOcean

LXD

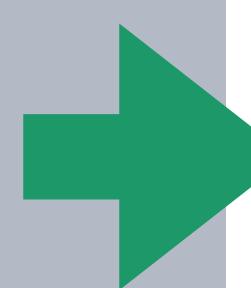
Parallels

VMware

Docker

NAVER Cloud

ProfitBricks



```
{  
  "variables": {  
    "aws_source_ami": "ami-bd8f33c5"  
  },  
  "builders": [  
    {  
      "type": "amazon-ebs",  
      "source_ami": "{{user `aws_source_ami`}}",  
      "ami_name": "geoffrey-tmp-{{timestamp}}",  
      "tags": {  
        "Created-by": "Packer",  
        "OS_Version": "Ubuntu",  
        "Release": "Latest"  
      }  
    }  
  ],  
  "provisioners": [  
    {  
      "type": "shell",  
      "inline": [  
        "git clone https://github.com/hashicorp/demo-terraform-101.git",  
        "sudo sh setup-web.sh"  
      ]  
    }  
  ]  
}
```

```
$ packer validate web.json
```

```
Failed to parse template: Error parsing JSON: invalid
character ':' after top-level value
At line 2, column 15 (offset 15):
```

```
 1:
 2:   "variables":
```

```
^
```

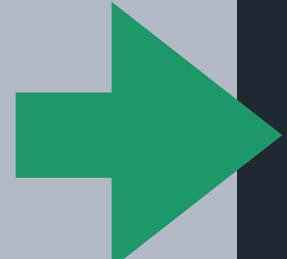
```
$ packer validate web.json
```

```
Template validated successfully.
```

```
$ packer build web.json

==> amazon-ebs: Waiting for the instance to stop...
==> amazon-ebs: Creating the AMI: geoffrey-tmp-1527132492
    amazon-ebs: AMI: ami-94334aec
==> amazon-ebs: Waiting for AMI to become ready...
==> amazon-ebs: Adding tags to AMI (ami-94334aec)...
==> amazon-ebs: Tagging snapshot: snap-0a604018916f9a8ec
==> amazon-ebs: Creating AMI tags
    amazon-ebs: Adding tag: "Created-by": "Packer"
    amazon-ebs: Adding tag: "OS_Version": "Ubuntu"
    amazon-ebs: Adding tag: "Release": "Latest"
==> amazon-ebs: Creating snapshot tags
==> amazon-ebs: Terminating the source AWS instance...
==> amazon-ebs: Cleaning up any extra volumes...
==> amazon-ebs: No volumes to clean up, skipping
==> amazon-ebs: Deleting temporary security group...
==> amazon-ebs: Deleting temporary keypair...
Build 'amazon-ebs' finished.

==> Builds finished. The artifacts of successful builds are:
--> amazon-ebs: AMIs were created:
us-west-2: ami-94334aec
```



Workflow for Developing Machine Images

Launch an instance with your desired base AMI

Write a script to install packages, etc.

Run the script and debug it on the live machine.

Copy the script to your local machine or to a repository that can be cloned.

Run the script from Packer with a `script` attribute.
(Do not try to run dozens of individual inline commands explicitly inside Packer).

Know Your OS

Lean on `systemd` or your OS-provided process launching and running facility.

Use OS-native commands to install packages rather than building from source.

Reboot the machine a few times to ensure all services start automatically.

For a simple example, see:

<https://github.com/hashicorp/demo-terraform-101/tree/master/assets>

EXERCISE

Validate and Build an Image With Packer

Using an existing Ubuntu image, write a Packer JSON file that clones a Git repo and runs a setup script contained within.

The setup script will copy a Go web application to the appropriate location. It will also install systemd startup scripts so the application is started on boot.

Multi-Provider

What You'll Learn

- ☑ How to configure multiple providers
- ☑ How to use attributes of one provider as arguments in another

Providers

Alicloud	Archive	AWS	1&1	OpenStack	OpenTelekomC
Azure	Bitbucket	CenturyLinkCloud	OpsGenie	Oracle Public Cloud	Oracle Cloud P
Chef	Circonus	Cloudflare	OVH	Packet	PagerDuty
CloudScale.ch	CloudStack	Cobbler	Palo Alto Networks	PostgreSQL	PowerDNS
Consul	Datadog	DigitalOcean	ProfitBricks	RabbitMQ	Rancher
DNS	DNSMadeEasy	DNSSimple	Random	Rundeck	Runscope
Docker	Dyn	External	Scaleway	SoftLayer	StatusCake
Fastly	GitHub	Gitlab	Spotinst	Template	Terraform
Google Cloud	Grafana	Heroku	Terraform Enterprise	TLS	Triton
HTTP	Icinga2	Ignition	UltraDNS	Vault	VMware vCloud
InfluxDB	Kubernetes	Librato	VMware NSX-T	VMware vSphere	
Local	Logentries	LogicMonitor			
Mailgun	MySQL	New Relic			
Nomad	NS1	Null			

```
provider "github" {
    token        = "${var.github_token}"
    organization = "hashicorp"
}

provider "aws" {
    version = ">= 1.19.0"
}

data "github_ip_ranges" "test" {}

egress {
    from_port    = 0
    to_port      = 0
    protocol     = "-1"
    cidr_blocks = ["${data.github_ip_ranges.test.pages}"]
}
```

EXERCISE

Use GitHub IP Addresses to Constrain an AWS Security Group

Configure a GitHub provider and an AWS provider.

Define a data source for GitHub IP address ranges.

Configure an AWS security group so that egress traffic is only allowed to access GitHub IP address ranges.

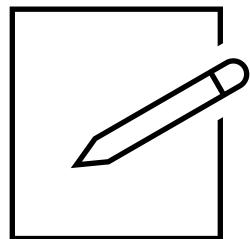
Verify by sending a ping to a GitHub IP address as well as to hashicorp.com

Resource Lifecycles

What You'll Learn

- ☑ How Terraform creates and destroys resources
- ☑ How to diverge from the default behavior
- ☑ What code organization techniques go with which lifecycle behavior

Default Behavior

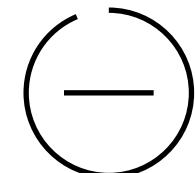


MODIFY

When possible

EXAMPLES

Add a tag
Change SSH user (GCP)

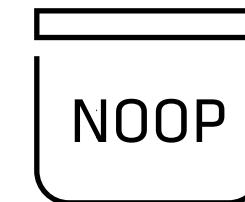


DESTROY

Freely

EXAMPLES

Rename an instance
Change machine type
Decrease count
Refactor to modules
Change startup script



Do Nothing

Occasionally

EXAMPLES

Change provisioner
Add an output
Refactor to variables

```
$ terraform plan
```

An execution plan has been generated and is shown below.
Resource actions are indicated with the following
symbols:

- ~ update in-place
- /+ destroy and then create replacement

Terraform will perform the following actions:

```
~ aws_instance.example
  vpc_security_group_ids.#:           "" =>
<computed>

-/+ aws_security_group.training (new resource required)
  id:                                "sg-6d36fc1c"
=> <computed> (forces new resource)
```

WARNING

Destroy, Then Create

Existing resources are first destroyed before new ones are created.

If the destruction succeeded cleanly, then and only then are replacement resources created.

```
$ terraform apply
```

```
aws_security_group: Still destroying... (14m00s elapsed)
aws_security_group: Still destroying... (14m10s elapsed)
```

Error: Error applying plan:

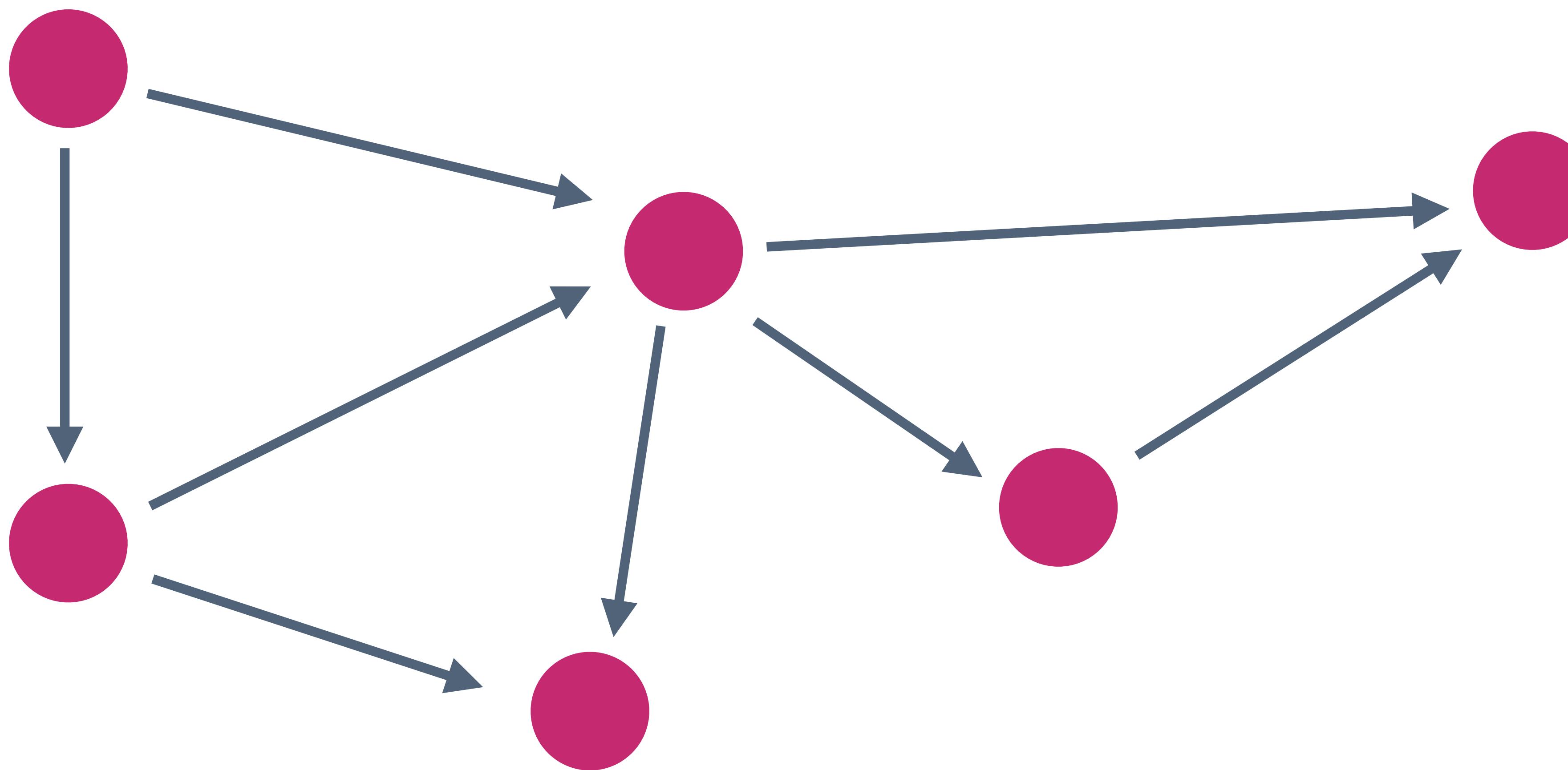
1 error(s) occurred:

```
* aws_security_group.training (destroy): 1 error(s) occurred:
```

```
* aws_security_group.training: DependencyViolation: resource sg-6d36fc1c has a dependent object status code: 400, request id: 4665247f-165b-46fc-b8ea-9c01a23dd4e9
```

```
lifecycle {  
  create_before_destroy = true  
  prevent_destroy = true  
}
```

Improved Application of Graph Theory



```
# Within an instance stanza
lifecycle {
    create_before_destroy = true
}
```



DASHBOARD

ACTIVITY

FILTER

Today

5:59 PM	 Completed: Create VM	302202337539-compute@developer.gserviceaccount.com created test-geoffrey-powerful-ant
5:59 PM	 Create VM	302202337539-compute@developer.gserviceaccount.com created test-geoffrey-powerful-ant
5:58 PM	 Completed: Delete VM	302202337539-compute@developer.gserviceaccount.com deleted test-geoffrey-known-hamster
5:58 PM	 Delete VM	302202337539-compute@developer.gserviceaccount.com deleted test-geoffrey-known-hamster

 Compute Engine
Products & services

VM instances



SHOW INFO PANEL

VM instances

Instance groups

Instance templates

Disks

Snapshots

Images

Filter VM instances



Columns ▾

<input type="checkbox"/> Name ^	Zone	Recommendation	Internal IP	External IP	Conn
test-geoffrey-careful-salmon	us-west1-b		10.138.0.3	35.185.207.173	SSH
test-geoffrey-powerful-ant	us-west1-b		10.138.0.2	35.197.126.186	SSH



DASHBOARD

ACTIVITY

FILTER

Today

- | | | |
|---------|----------------------|---|
| 6:04 PM | Completed: Delete VM | 302202337539-compute@developer.gserviceaccount.com deleted test-geoffrey-powerful-ant |
| 6:03 PM | Delete VM | 302202337539-compute@developer.gserviceaccount.com deleted test-geoffrey-powerful-ant |
| 6:03 PM | Completed: Create VM | 302202337539-compute@developer.gserviceaccount.com created test-geoffrey-careful-salmon |
| 6:03 PM | Create VM | 302202337539-compute@developer.gserviceaccount.com created test-geoffrey-careful-salmon |

```
# Within an instance stanza
provisioner "local-exec" {
  command = "curl --max-time 60 http://${self.public_ip}"
}
```

Two local-exec Tips

Within a provisioner, use `self.ATTRIBUTE` to access values that would usually be accessed from outputs. For example, `self.public_ip`

Know your health check tool. For example, curl will return error (`$? > 0`) if no server is found, but won't return error if a 404 page is served.

*Does this resource need to be
managed by this project?*

WARNING

Plan Does Not Reflect Lifecycle Directives

Even while using `create_before_destroy`, the output of `terraform plan` will describe the default (destroy first).

`-/+` destroy and then create replacement

Partial Applies Will Clean Up On The Next Apply

When using `create_before_destroy`, it's possible to encounter a runtime error where Terraform cannot complete the apply.

This could leave both the old and the new instances intact.

The old instance is stored in `terraform.tfstate` under a `deposed` key. It will be destroyed after the next successful apply.

Prevent Destroy

Warns if any change would result in destroying a resource

All resources that this resource depends on must also be set to `prevent_destroy`

```
lifecycle {  
    prevent_destroy = true  
}
```

```
Error: Error running plan: 1 error(s) occurred:  
  
* google_compute_instance.test: 1 error(s) occurred:  
  
* google_compute_instance.test:  
google_compute_instance.test: the plan would destroy this  
resource, but it currently has lifecycle.prevent_destroy  
set to true. To avoid this error and continue with the  
plan, either disable lifecycle.prevent_destroy or adjust  
the scope of the plan using the -target flag.
```

TIP

Failure is the Only Option!

Some practitioners want the ability to see a warning but continue anyway.

That workflow is not supported. If an error is encountered, the rest of the execution will stop until it is resolved.

Sentinel

Agenda

Why Sentinel?

Syntax

Data structure

Debugging

Common use cases

Workflow + API

Testing

Deployment

Architecture

Why Write Policy as Code with Sentinel?

Sandboxing

Codification

Version Control

Testing

Automation

What Sentinel Does

Policy as code

Run between plan and apply

Analyzes current state plus
values that can be calculated
before apply

Analyzes state of configuration

Analyzes state of system at
deploy time

Multi-cloud, multi-provider

Should be a piece of your
deployment security strategy,
together with IAM policies

What Sentinel Does Not Do

Continuous security checks

Audit previously deployed systems

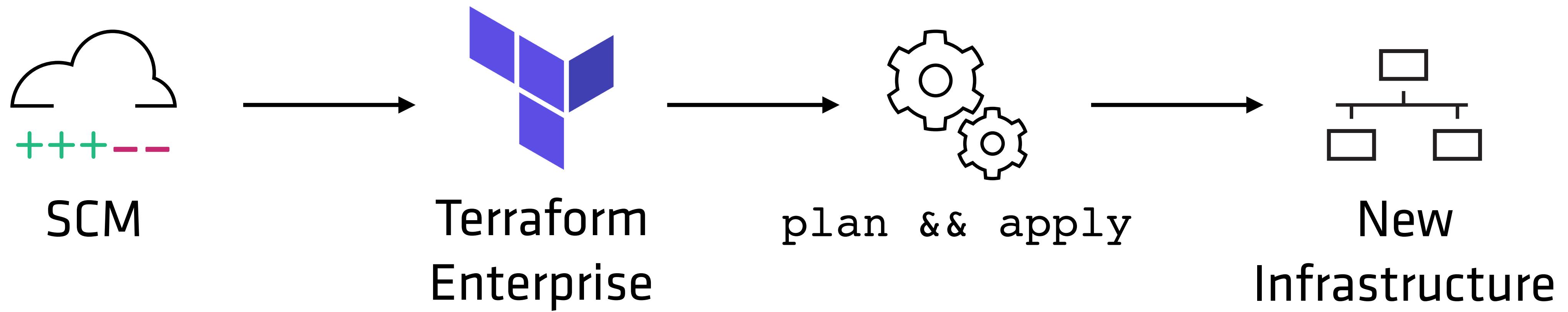
Analyze executable contents of a VM/container/etc.

Limit runtime actions of deployed applications

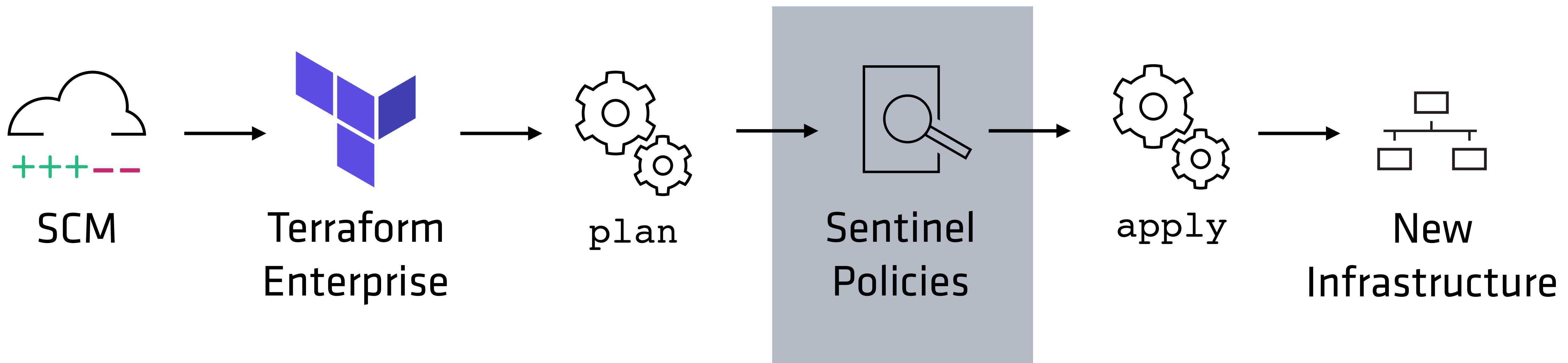
*In its current implementation,
Sentinel is a tool for preventing
mistakes by operators acting in
good faith.*

- Martin Atkins (Terraform Tech Lead)

How it works without Sentinel



How it works with Sentinel



Become Familiar With All Sentinel Documentation

Main Sentinel docs:

<https://docs.hashicorp.com/sentinel/>

Terraform Enterprise docs:

<https://www.terraform.io/docs/enterprise/sentinel/index.html>

Blog articles:

<https://www.hashicorp.com/blog/category/sentinel>

Your First Policy

Download Sentinel Simulator

Download Sentinel Simulator

Sentinel Simulator enables you to develop and test Sentinel policies locally on your own machine or in a CI environment. It can also be used as a testing tool for [writing custom imports](#).

Below are the available downloads for the latest version of Sentinel Simulator (0.1.0). Please download the proper package for your operating system and architecture.

You can find the [SHA256 checksums](#) for Sentinel Simulator 0.1.0 online and you can [verify the checksums signature](#) file which has been signed using [HashiCorp's GPG key](#). You can also [download older versions](#) of Sentinel Simulator from the releases service.

 **Mac OS X**
[32-bit](#) | [64-bit](#)

 **FreeBSD**
[32-bit](#) | [64-bit](#) | [Arm](#)

EXERCISE

Run Task 1 in Lab

Download Sentinel Simulator, copy to \$PATH

Try a simple policy named simple.sentinel:

```
main = rule { true }
```

Syntax

What you'll learn

 Basic syntax

 Rules, variables, loops, functions, conditionals

```
main = rule { true }
```

```
# Conditional rule
# Is true if is_docker is false
# Otherwise examines has_exposed_ports
main = rule when is_docker { has_exposed_ports }
```

```
==      equal
!=      not equal
<      less
<=     less or equal
>      greater
>=     greater or equal
"is"    equal
"is not" not equal
```

TIP

Use as Precise a Comparator as You Can

For equality, use `is` rather than `contains`

Boolean Values

undefined or true	= true
undefined or false	= undefined
undefined or undefined	= undefined
undefined and true	= undefined
undefined and false	= undefined
undefined and undefined	= undefined
undefined xor true	= undefined
undefined xor false	= undefined
undefined xor undefined	= undefined

Rules Are Aggressively Memoized

Rules are lazy and memoized (will not be recomputed unless absolutely necessary).

```
rule when Y { Z == true }
```

Write small rules

It's slightly easier to debug policies if the rules are short and there are many of them.

```
main = rule {
    bacon_is_eggs and
    pancake_is_maple and
    has_id
}
```

Collections

all collection as index, item { } # Returns bool

any ... # Returns bool

for ... # NOTE: Does not return

contains

matches

WARNING

Lists Need an Iterator

There is sometimes no way to access a value except with a loop (such as `tfplan.animals.0.id`)

Functions

```
a = func(x) { return x*2 }
```

Must return (or return undefined)

Testing

Understanding Sentinel Tests

Terraform code is the implementation (outside your Sentinel project)

The state of your infrastructure is the result, but it needs to be verified against the policy before it is built. So only the JSON output of the plan is tested.

The policy itself is the test.

A JSON data file is both the fixture and the expectation.

```
main = rule { true }
```

```
└── time.sentinel
└── test
    └── time
        ├── fail.json
        └── pass.json
```

```
{  
  "test": {  
    "main": true  
  }  
}
```

```
{  
  "test": {  
    "main": true  
  },  
  "mock": {  
    "timespace": {  
      "hour": 10,  
      "weekday": 2  
    }  
  }  
}
```

```
{  
  "test": {  
    "main": true  
  },  
  "global": {  
    "environment": "staging"  
  }  
}
```

```
$ sentinel --help test
```

Test cases are expected to be in "test/<policy>/*.json" files where "<policy>" is the name of the policy filename without an extension.

```
$ sentinel test
```

```
PASS - with-data.sentinel
```

```
PASS - test/with-data/expected-failure.json  PASS -  
test/with-data/pass.json  PASS - test/with-data/embedded-  
values.json
```

EXERCISE

Implement Task 2 in Lab

Create a Sentinel policy that works with slightly more real-world data structures.

WARNING

Errors are minimal

You might see that a test fails but not have a clear reason, even with --verbose

For example, doing `import "tfplan"` in a policy will cause the test to fail unless mocked (because it can't get to the import).

Within TFE

Terraform Enterprise | geoffrey X +

https://app.terraform.io/app/geoffrey-org

geoffrey-org Workspaces Modules Documentation | Status

Workspaces ORGANIZATION geoffrey-org ✓

All (1) Organization Settings

WORKSPACES SWITCH ORGANIZATIONS topfunky-demo > Create new organization

+ New Workspace

Needs Attention (0) Running (0) Filter Sort Search Workspaces by name

TUS	LATEST CHANGE	RUN	REPO
ED✓	a month ago	run-jkhg	geoffrey-hashicorp/training-lab

Create new organization

HashiCorp

Terms Privacy Security © 2018 HashiCorp, Inc.

https://app.terraform.io/app/geoffrey-org/settings/profile

The screenshot shows a web browser window for Terraform Enterprise, specifically the organization settings page for 'geoffrey-org'. The URL in the address bar is <https://app.terraform.io/app/geoffrey-org/settings/profile>.

The left sidebar lists organization settings:

- Profile** (selected)
- Teams
- OAuth Configuration
- API Token
- User Sessions
- Manage SSH Keys
- Sentinel Policy

The main content area displays the **Organization Profile** for 'geoffrey-org' with the following details:

- NAME**: geoffrey-org
- NOTIFICATION EMAIL**: geoffrey@hashicorp.com

A large red warning box titled **Organization Delete** contains the message: "Deleting the **geoffrey-org** organization will permanently delete all workspaces associated with it. Please be certain you know what you're doing. This action cannot be undone." Below this is a red button labeled **Delete this organization**.

The bottom of the page shows the URL <https://app.terraform.io/app/geoffrey-org/settings/policies>.

The screenshot shows the Terraform Enterprise web interface. The top navigation bar includes the Terraform logo, the workspace name "geoffrey", and links for "Documentation" and "Status". The left sidebar, titled "ORGANIZATION SETTINGS", lists options like "Profile", "Teams", "OAuth Configuration", "API Token", "User Sessions", "Manage SSH Keys", and "Sentinel Policy", with "Sentinel Policy" currently selected. The main content area is titled "Sentinel Policy" and describes it as rules enforced on workspace runs to validate compliance. It lists four existing policies: "aws-tags" (updated December 18th 2017), "print" (updated November 29th 2017), "require-tag-bacon" (updated February 23rd 2018), and "learning-demo" (updated March 6th 2018), each with an "Edit" button.

Sentinel Policy

Sentinel Policies are rules which are enforced on every workspace run to validate the terraform plan and corresponding resources are in compliance with company policies.

aws-tags	Last updated December 18th 2017, 1:58 pm	Edit
print	Last updated November 29th 2017, 6:55 pm	Edit
require-tag-bacon	Last updated February 23rd 2018, 4:23 pm	Edit
learning-demo	Last updated March 6th 2018, 6:01 pm	Edit

Terraform Enterprise | geoffrey X +

https://app.terraform.io/app/geoffrey-org/settings/policies/pol-VxL17vqc4h4AwfQz/edit

Profile
Teams
OAuth Configuration
API Token
User Sessions
Manage SSH Keys
Sentinel Policy

POLICY NAME
aws-tags

The name of your policy is used in the UI and Sentinel output. Accepts alphanumeric characters, as well as - and _.
Cannot be modified after creation.

ENFORCEMENT MODE
soft-mandatory (can override)

Only users on the owners team in this organization can override **soft-mandatory** checks.

POLICY CODE

```
1 import "tfplan"
2
3 has_at_least_one_tag = rule {
4     all tfplan.resources.aws_instance as _, instances {
5         all instances as _, r {
6             (length(r.applied.tags) else 0) >= 1
7         }
8     }
9 }
10
11 main = rule {
12     has_at_least_one_tag
13 }
```

Save policy **Delete policy**

Enforcement Levels

hard-mandatory

soft-mandatory

advisory

```
"tfplan": {  
    "random_pet": {  
        "server": {  
            "0": {  
                "applied": {  
                    "id": "deciding-pegasus",  
                    "length": "2",  
                    "separator": "-"  
                },  
                "diff": {}  
            }  
        }  
    }  
}
```

Read Complex Examples on GitHub

HashiCorp staff frequently build and publish sample policies.

Find them in places such as the `terraform-guides` repository:

[https://github.com/hashicorp/terraform-guides/
blob/master/governance/aws/restrict-aws-
region.sentinel](https://github.com/hashicorp/terraform-guides/blob/master/governance/aws/restrict-aws-region.sentinel)

TIP

Read documentation on imports

Read, understand, and refer to the documentation on imports (tfplan, tfconfig, tfstate):

[https://www.terraform.io/docs/enterprise/sentinel/
import/index.html](https://www.terraform.io/docs/enterprise/sentinel/import/index.html)

EXERCISE

Task 3: Run a policy within Terraform Enterprise

Log into TFE

Link GitHub repo if it's not already there (demo-terraform-101)

Add policy

Make change on GitHub

View output of policy

Debugging

WARNING

app.terraform.io is not an IDE

Don't try to develop in the browser.

Work on the local machine, write tests. Work in small increments if possible.

Run a syntax check with `fmt` if nothing else is possible.

The cycle of write-execute-observe is long in TFE, especially if your core Terraform plan is large.

WARNING

Warnings are minimal

You won't be warned if you use an outer variable. No printout of what didn't match.

WARNING

Computed values can be loopholes

When possible, use a whitelist instead of a blacklist.

Example: no cidr 0.0.0.0/0 vs 192.168.0.0/16

Whitelists are more secure than blacklists, in general.

Example: 000.000.000.000/0 is a valid IP that could evade checks for 0.0.0.0/0

WARNING

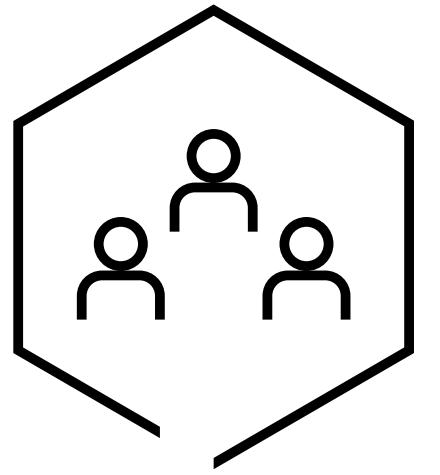
The applied data structure is most useful

When examining the `tfplan` data structure, look for `applied` to find the values that will be deployed

Authoring with the Registry

What You'll Learn

- ✓ Use standard module structure to build a module
- ✓ Publish a module to the repository
- ✓ Use the published module from another Terraform configuration
- ✓ Understand how to version and update a module
- ✓ Use the module configuration designer

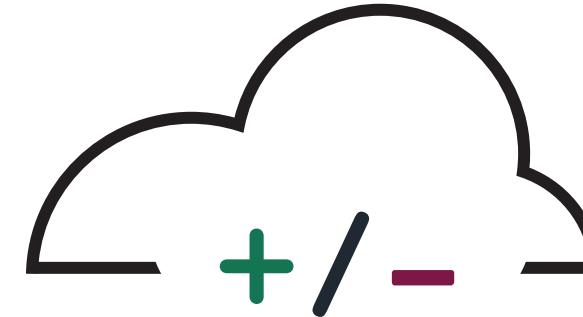


Modules

Collaboration, Reuse,
Consistency

EXAMPLES

AWS Networking
App Server
DB Cluster

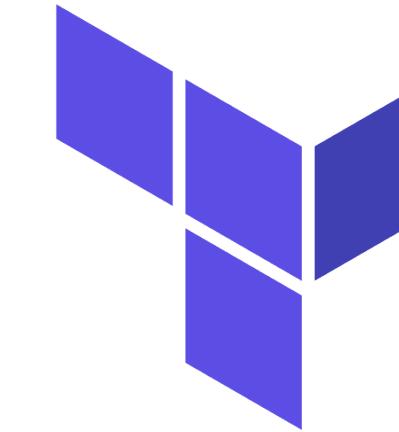


SCM

Access Control,
Pull Request Workflow, CI

EXAMPLES

GitHub
Bitbucket (private)



Terraform Registry

Versioning, Documentation,
Search

EXAMPLES

terraform-aws-networking
terraform-go-server
terraform-pg-cluster

Private Module Registry

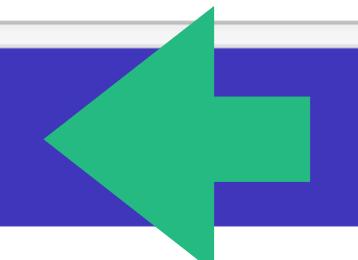
Both the hosted and private instances of Terraform include your own private module registry which can be used to securely and privately publish modules within your company.



geoffrey-org

Workspaces

Modules



Documentation | Status



Modules

+ Design Configuration

+ Add Module

consul



Providers

animal

PRIVATE

Sample repo for experimenting with the HashiCorp Terraform
Module Registry

Details

DEMO

Version 0.0.2

Module Registry Workflow: Git

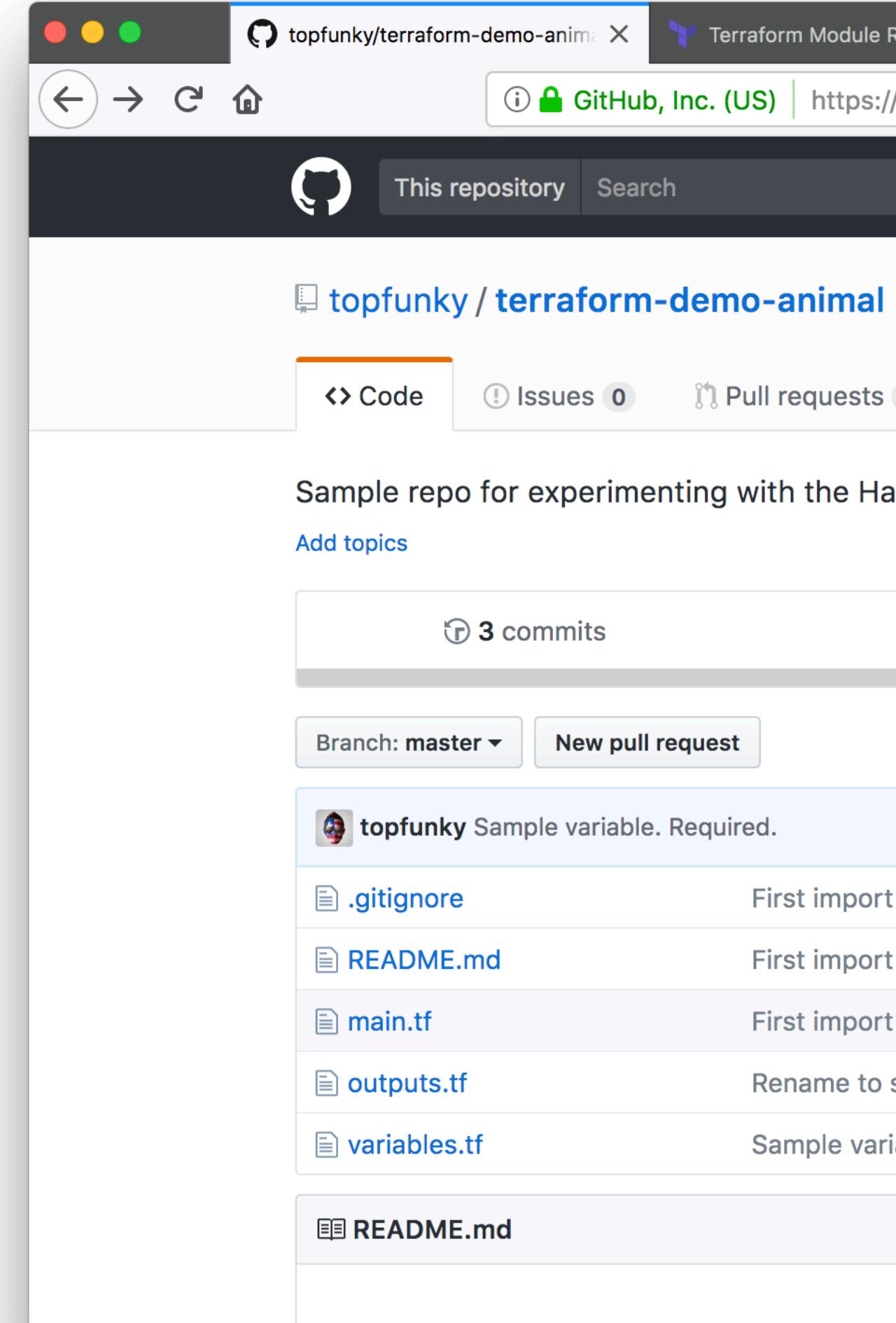
Push code to GitHub or Bitbucket. The repository must be named in the format:

terraform-PROVIDER-MODULE

Examples:

terraform-aws-peering

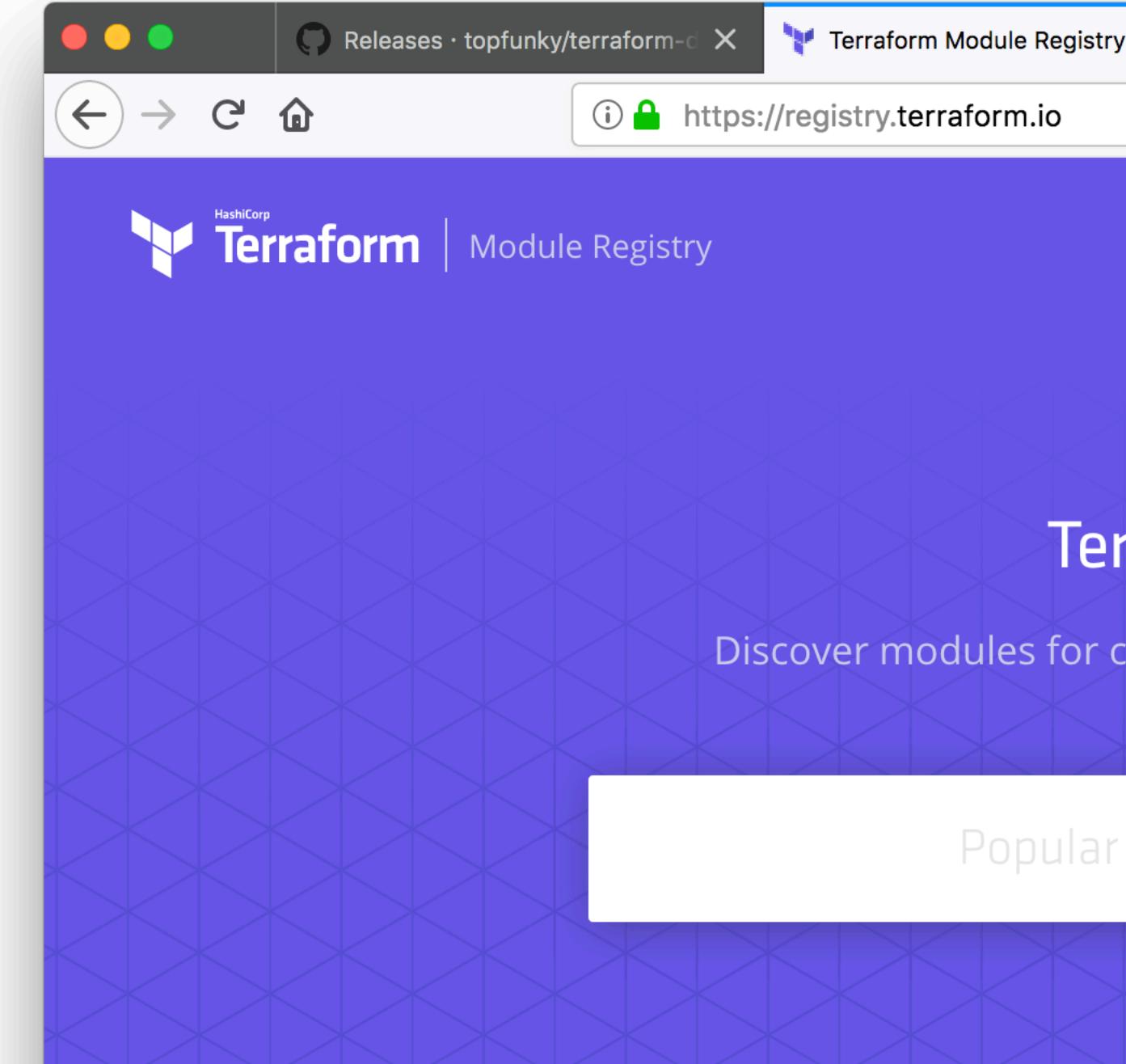
terraform-azure-autoscaling



Module Registry Workflow: Registry

Connect your SCM account to the Terraform Registry.
Publish a module. Tagged commits will be synced
automatically.

v1.0.6



Use and learn

Module Registry Workflow: Config

Use the code snippet from the registry to reference the module from your code. Provide all necessary variables.

Run `init` to pull in the correct version of the module.

```
terraform init
```

```
~/projects/terraform-demo-animal
.
├── README.md
├── LICENSE
├── main.tf
├── outputs.tf
└── variables.tf
```

```
# main.tf
resource "random_pet" "animal" {}

# outputs.tf
output "animal" {
    value      = "${random_pet.animal.id}"
    description = "Contains the name of a random animal."
}
```

EXERCISE

Fork a GitHub Repository and Tag Your Code

Fork the existing `hashicorp/terraform-demo-animal` public GitHub repository.

Tag a commit with a version number and push that as well:

```
git tag v0.0.1
```

```
git push --tags
```

A screenshot of a GitHub repository page for 'hashicorp/terraform-demo-animal'. The page shows basic repository statistics: 3 commits, 1 branch, 0 releases, 1 contributor, and MIT license. A large blue arrow points to the 'Fork' button in the top right corner of the header.

hashicorp/terraform-demo-animal

Demo code as part of the HashiCorp Terraform Enterprise 201 course. This code is used to demonstrate the public and private module registries. <https://www.hashicorp.com/training>

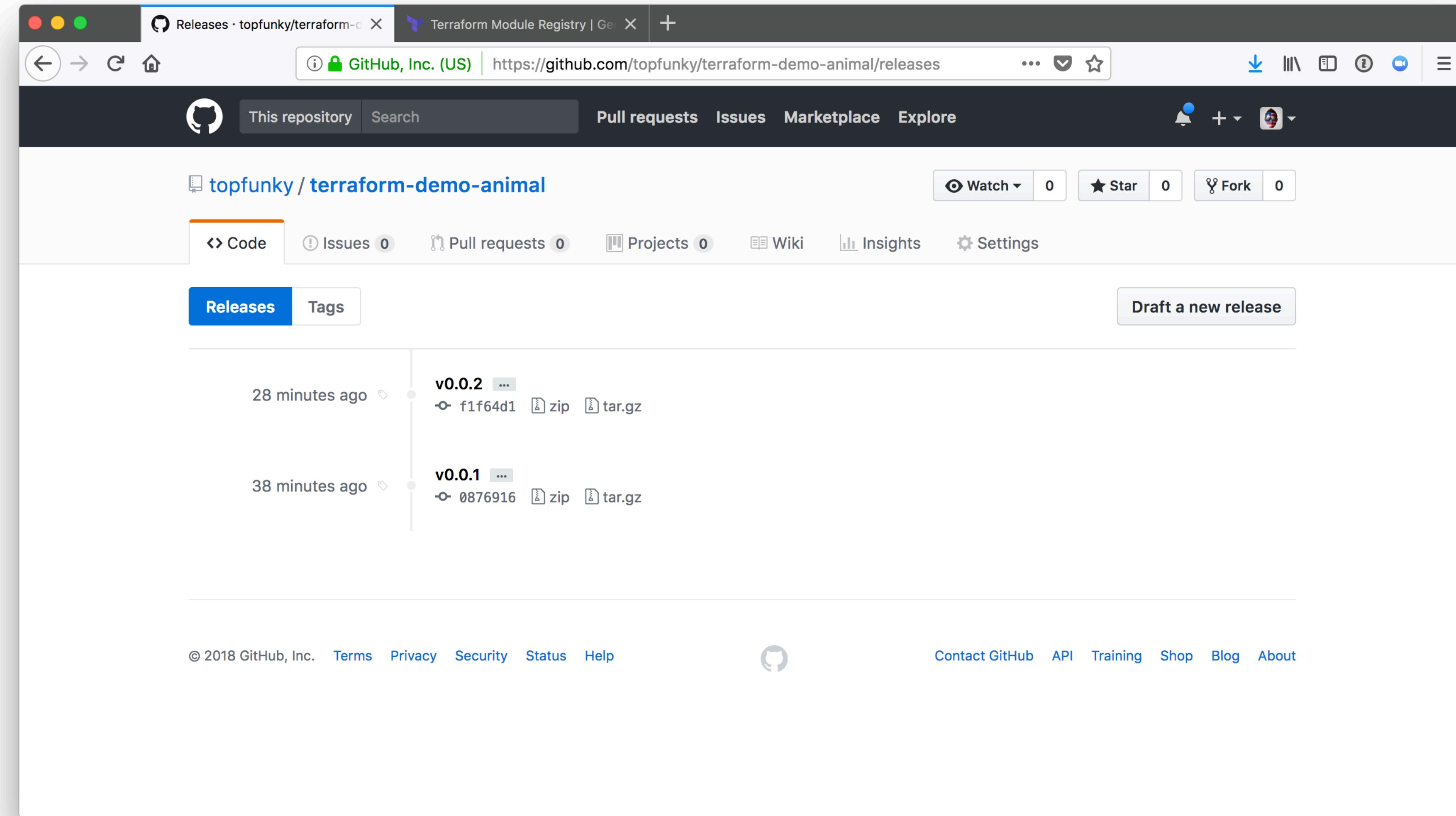
Add topics

Branch: master ▾ New pull request

Create new file Upload files Find file Clone or download ▾

File	Description	Last Commit
.gitignore	Initial commit	5 minutes ago
LICENSE	Initial commit	5 minutes ago
README.md	Description of demo project.	2 minutes ago
main.tf	Implementation code for module demo.	2 minutes ago
outputs.tf	Implementation code for module demo.	2 minutes ago
variables.tf	Implementation code for module demo.	2 minutes ago
README.md		

```
$ git clone https://github.com/$USER/terraform-demo-animal.git  
  
$ cd terraform-demo-animal  
  
$ git tag v0.0.1  
  
$ git push --tags
```



EXERCISE

Publish the Code on Your Private Module Registry

Go to <https://app.terraform.io/> and go to the workspaces dashboard.

Click the "Modules" button in the menu bar.

Use the "+ Add Module" button to select your repository and add it to the module registry.

```
# main.tf
provider "random" {
  version = "~> 1.1"
}
```

```
$ tree .terraform

.terraform
└── modules
    ├── 495f1ee96941ccf0a6c619930ea1160d
    │   └── topfunky-terraform-demo-animal-f1f64d1
    │       ├── README.md
    │       ├── main.tf
    │       ├── outputs.tf
    │       └── variables.tf
    ├── 753525cdd62578ec39462ba136cf9f7e
    │   └── topfunky-terraform-demo-animal-0876916
    │       ├── README.md
    │       ├── main.tf
    │       └── output.tf
    └── modules.json

└── plugins
    └── darwin_amd64
        ├── lock.json
        └── terraform-provider-random_v1.1.0_x4
```

```
{  
  "Modules" : [  
    {  
      "Root" : "topfunky-terraform-demo-animal-0876916",  
      "Version" : "0.0.1",  
      "Source" : "topfunky/animal/demo",  
      "Key" : "1.animal;topfunky/animal/demo.0.0.1",  
      "Dir" : ".terraform/modules/753525cdd62578ec39462ba136cf9f7e"  
    },  
    {  
      "Dir" : ".terraform/modules/495f1ee96941ccf0a6c619930ea1160d",  
      "Source" : "topfunky/animal/demo",  
      "Key" : "1.animal;topfunky/animal/demo.0.0.2",  
      "Version" : "0.0.2",  
      "Root" : "topfunky-terraform-demo-animal-f1f64d1"  
    }  
  ]  
}
```

TIP

By Default, Paths are Relative to the Terraform CLI

If you want to reference resources in a published module's own directory, use

```
 ${path.module}
```

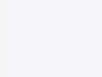
Module Configuration Tool

Module Configuration Tool

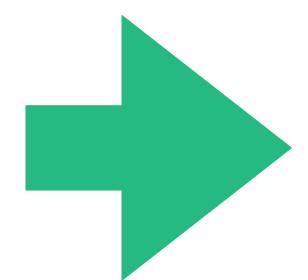
User interface for selecting modules and specifying required values

Code generator (write only, no editing)

Explore APIs quickly



Modules

[+ Design Configuration](#)[+ Add Module](#)

consul

[Providers ▾](#)

animal

PRIVATE

Sample repo for experimenting with the HashiCorp Terraform
Module Registry

[Details](#)

DEMO

Version 0.0.2

Terraform Enterprise | Modules X +

geoffrey-org Workspaces Modules Documentation Status

Modules / Configuration Designer

Select Modules > Set Variables > Verify > Publish Next »

consul Providers

ADD MODULES TO WORKSPACE

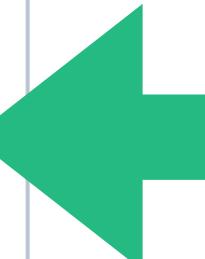
animal PRIVATE

Sample repo for experimenting with the HashiCorp Terraform Module Registry

Details Add Module

DEMO Version 0.0.2

SELECTED MODULES 0



Terraform Enterprise | Modules X +

geoffrey-org Workspaces Modules Documentation Status

Modules / Configuration Designer

Select Modules > Set Variables > Verify > Publish

Next »

consul

Providers

ADD MODULES TO WORKSPACE

animal PRIVATE Version 0.0.2

Sample repo for experimenting with the HashiCorp Terraform Module Registry

Details ↗ Add Module

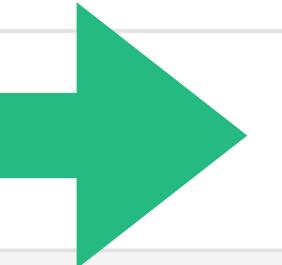
DEMO Version 0.0.2

SELECTED MODULES 1

animal PRIVATE Version 0.0.2

Sample repo for experimenting with the HashiCorp Terraform Module Registry

Details ↗ Remove ✕





Modules / Configuration Designer

Select Modules > Set Variables > Verify > Publish

Next »

SELECT MODULE TO CONFIGURE

animal PRIVATE

Sample repo for experimenting with the HashiCorp Terraform Module Registry

[Details ↗](#)[Configured ✓](#)

CONFIGURE VARIABLES

name REQUIRED

a name

server

 Deferred

The screenshot shows the Terraform Enterprise Modules interface. At the top, there's a header bar with the Terraform logo, the text "Terraform Enterprise | Modules", and a "+" button. Below the header is a navigation bar with back, forward, and home icons, followed by a URL bar showing "https://app.terraform.io/app/modules/geoffrey-org/design/verify". To the right of the URL are three dots, a star icon, and several small blue icons. The main menu bar has a user icon, "geoffrey-org", "Workspaces", and "Modules" (which is highlighted in blue). On the far right of the menu bar are "Documentation", "Status", and a user profile icon.

Modules / Configuration Designer

Select Modules > Set Variables > **Verify** > Publish

Next »

Terraform Configuration

```
1 //-----
2 // Modules
3 module "animal" {
4   source  = "app.terraform.io/geoffrey-org/animal/demo"
5   version = "0.0.2"
6
7   name = "server"
8 }
```

Module Configuration Designer

Supports the producer/consumer pattern.

Teams can create well-documented modules with clear arguments that can be configured and used by other teams. Only minimal coding ability is required.

Producer teams can define best practices in code. Consuming teams don't need to understand the details, but can comply with infrastructure provisioning and security policies.

What You Learned

- ✓ Use standard module structure to build a module
- ✓ Publish a module to the repository
- ✓ Use the published module from another Terraform configuration
- ✓ Understand how to version and update a module
- ✓ Use the module configuration designer