



Scaling Terraform at ThousandEyes

SREcon23 Americas

Ricard Bejarano

Site Reliability Engineer, Infrastructure

The problem

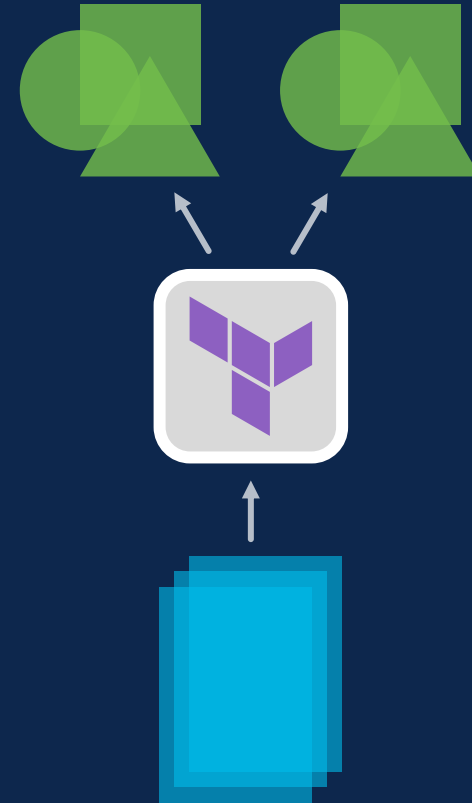
The problem

Single Terraform deployment



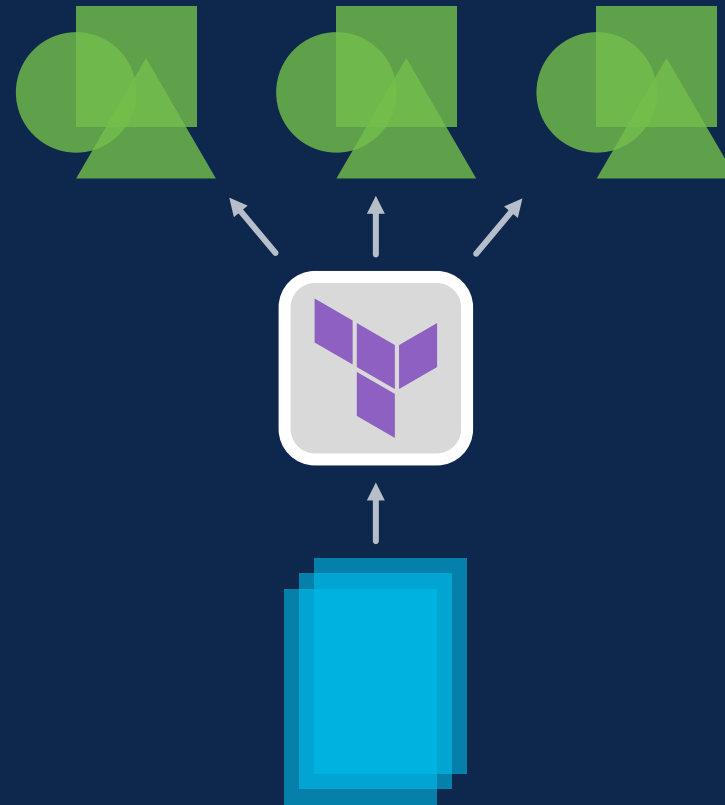
The problem

Let's add staging



The problem

Now scale
that up



Plan times

Any **single change** triggers a plan

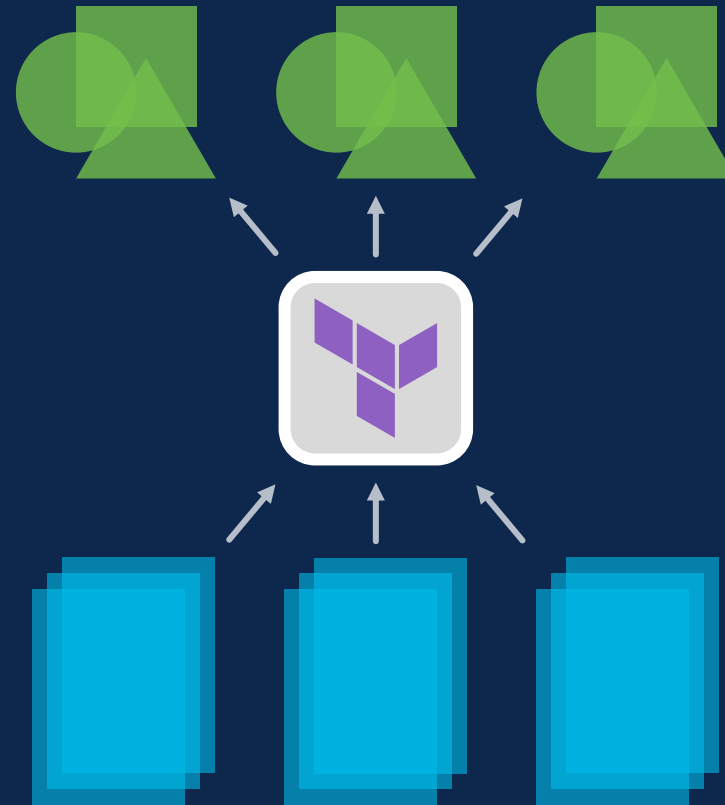
Terraform refreshes **every resource**

No, you **can't really skip** refresh



The problem

So we **split**
into separate
deployments



Drift potential

Separate deployments means **different files**

Differences across those files is **drift**

Reconciling drift in Terraform is **awful**



The problem

Boilerplate

Every deployment does **the same**

Backends, providers, variables, etc.

Those are all just **boilerplate**



Plan times

If we put our entire infrastructure in a single Terraform deployment, **plans take forever.**

So we **split our infrastructure** into **multiple deployments.**

Drift potential

By splitting, we **allow for drift** to creep into our configuration.

Fixing Terraform drift is **horrible.**

Boilerplate

Additionally, we reinitialize the same state backend, providers, variables, etc.

This materialized as **duplicate, non-functional code.**



This doesn't scale



Our solution

We looked into Terragrunt

We already had 5000+ Terraform files

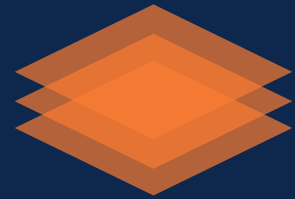
Required **learning** a new tool

Didn't solve the **boilerplate** problem



So we built our own



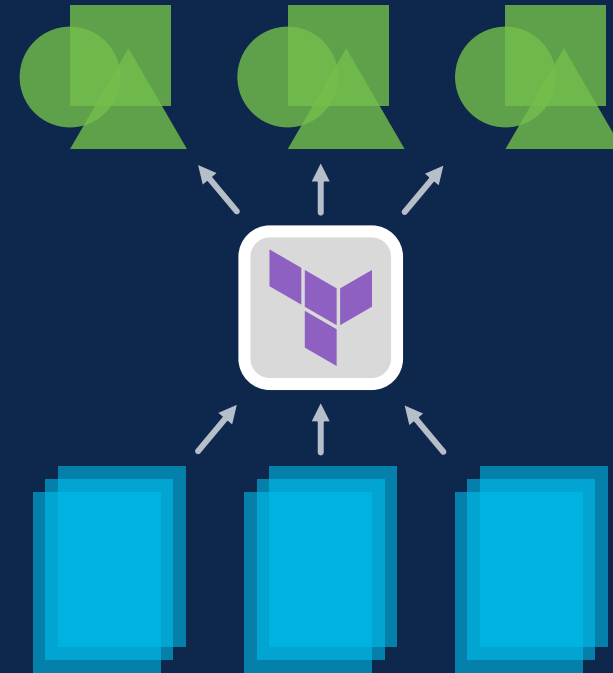


Stacks for Terraform



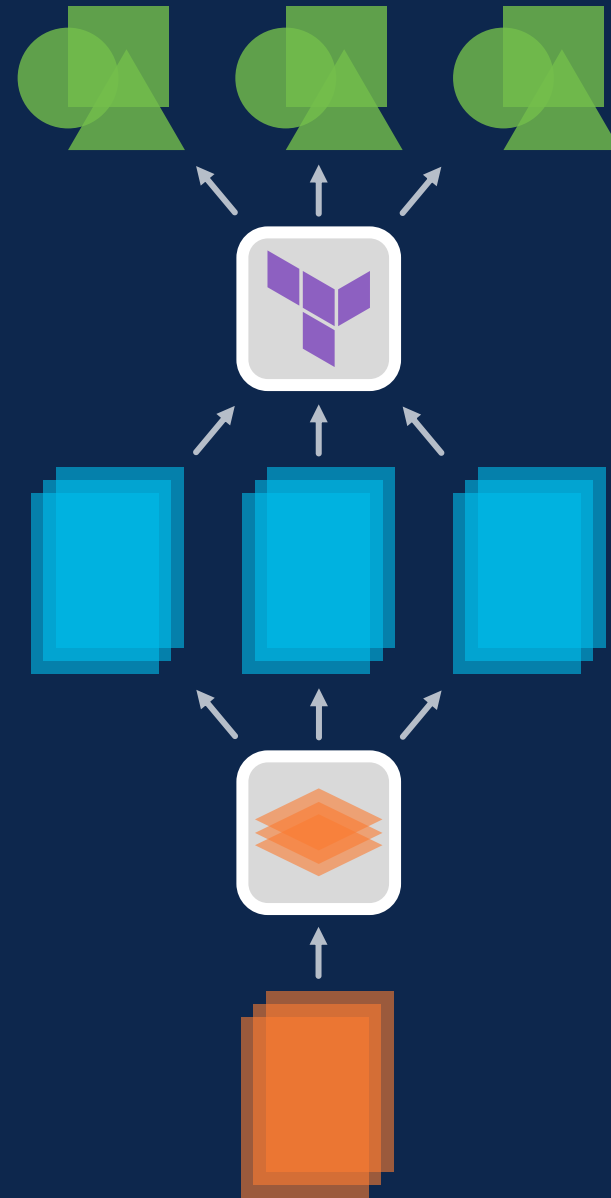
Our solution

Remember this?

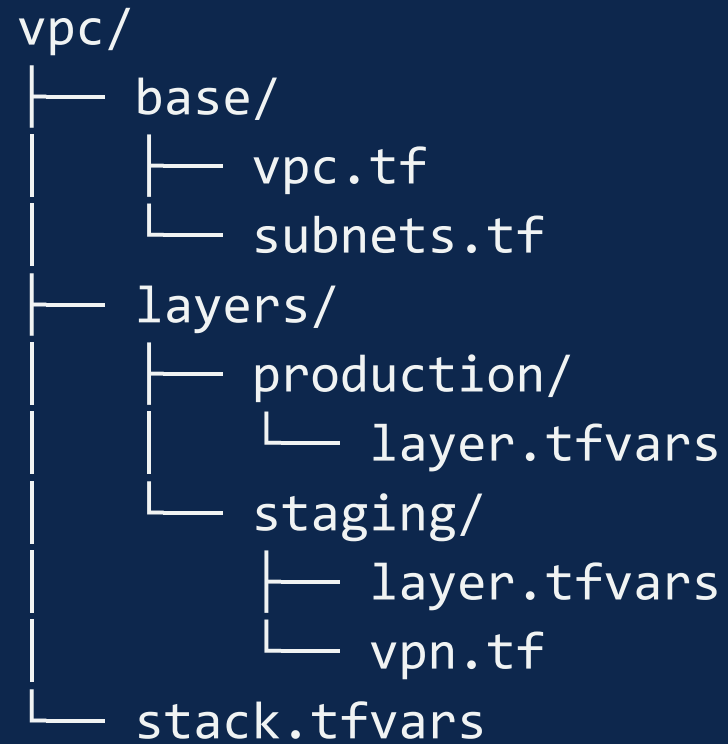


Our solution

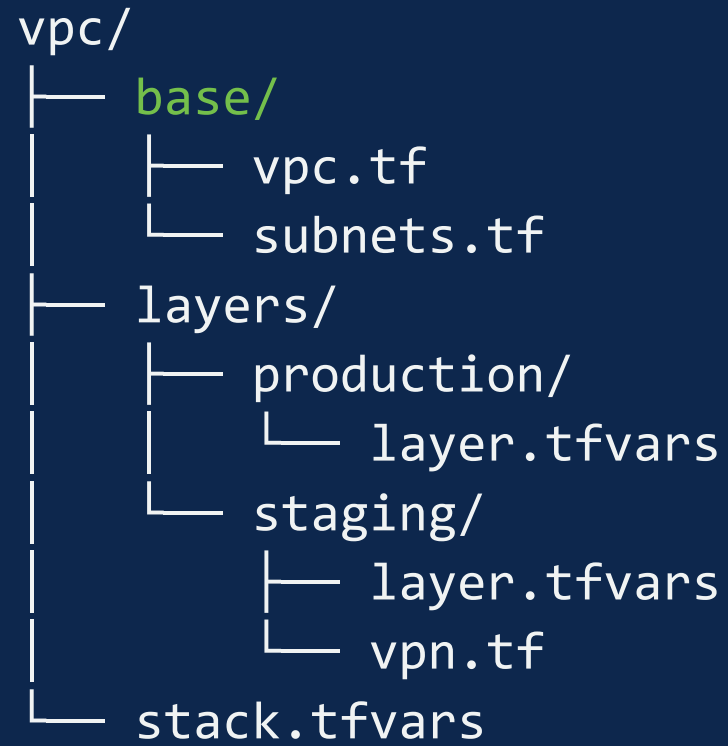
Stacks for Terraform



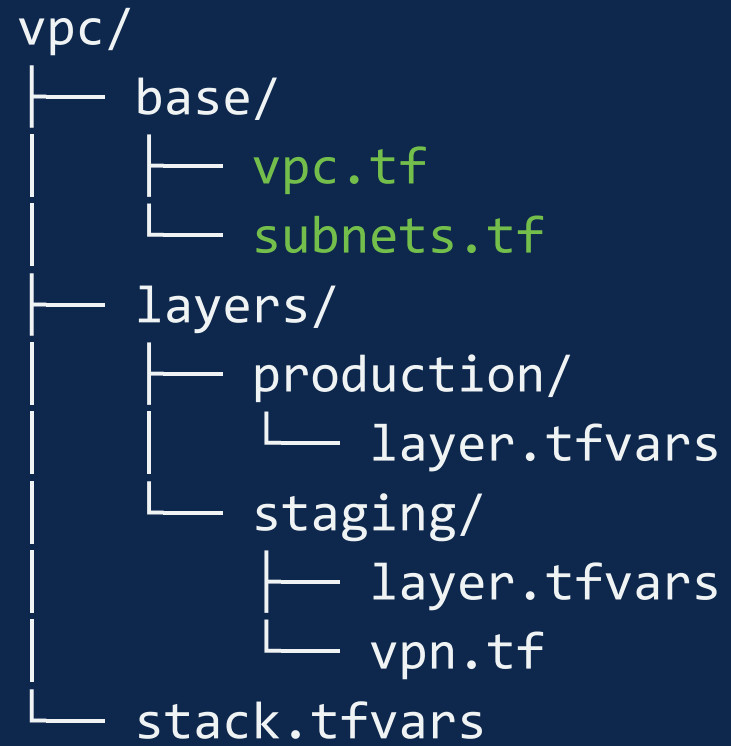
Example



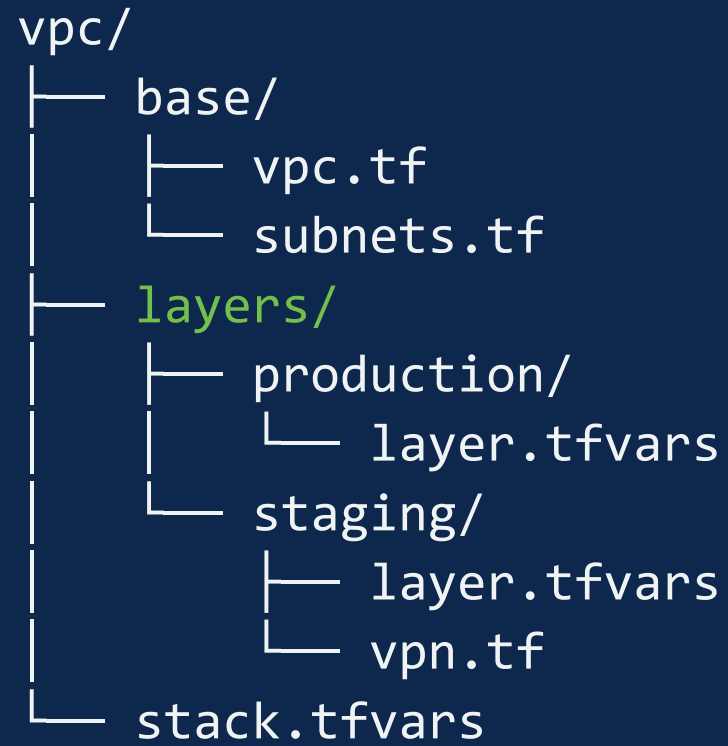
Example



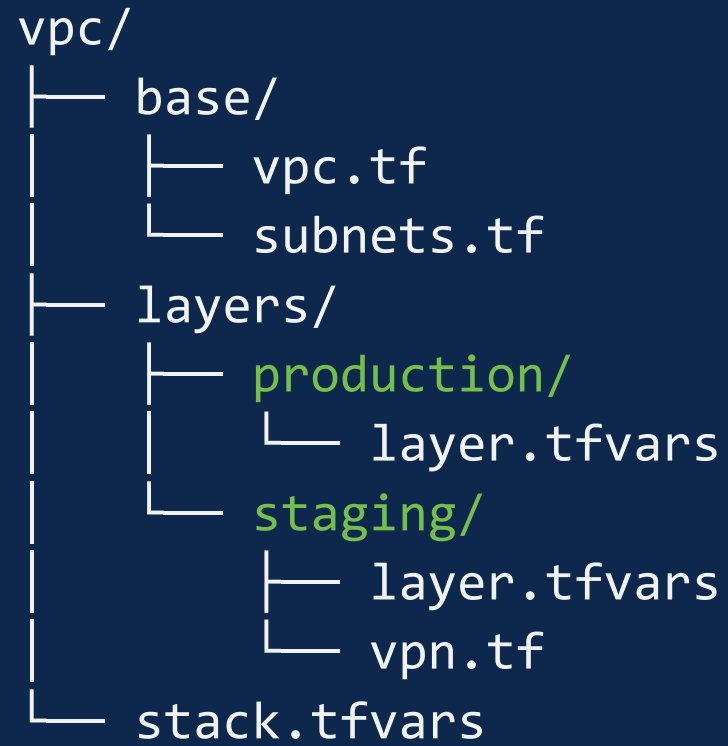
Example



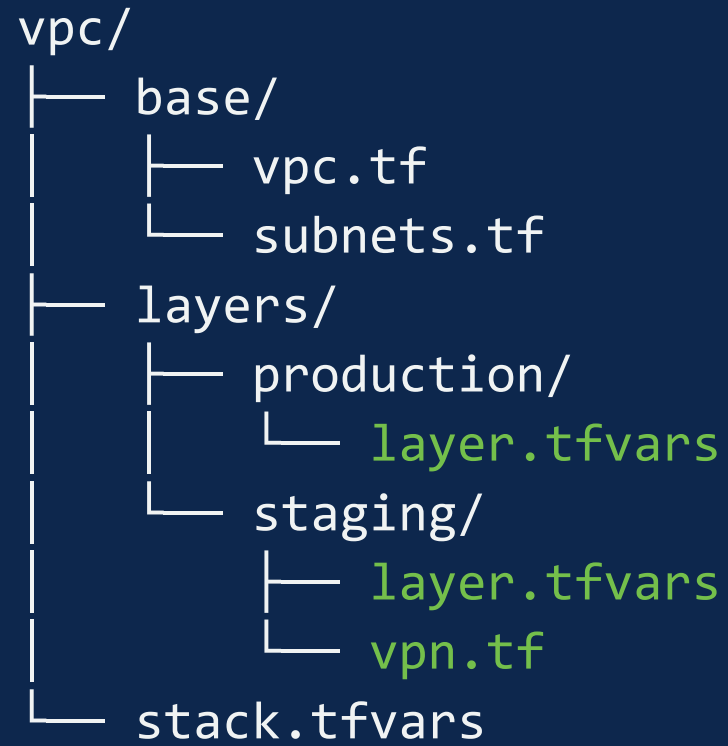
Example



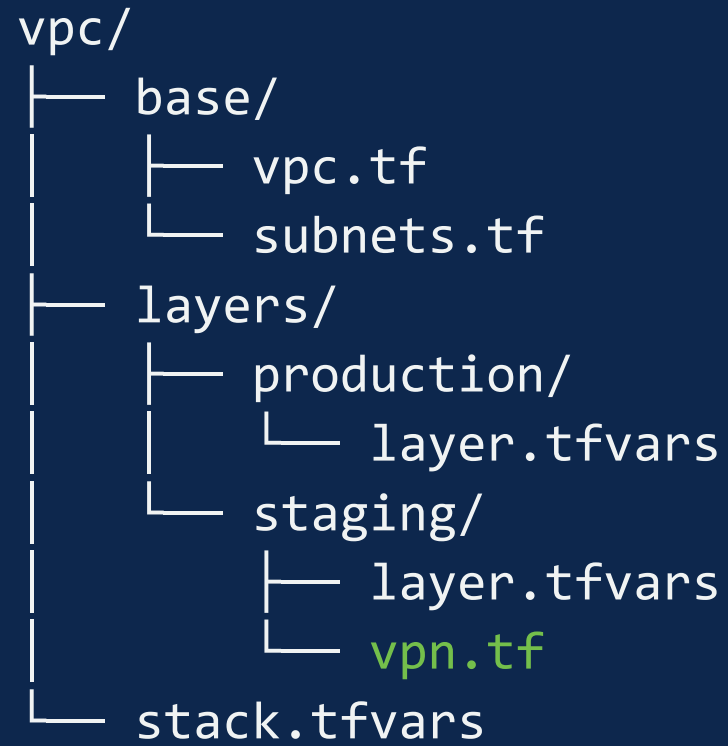
Example



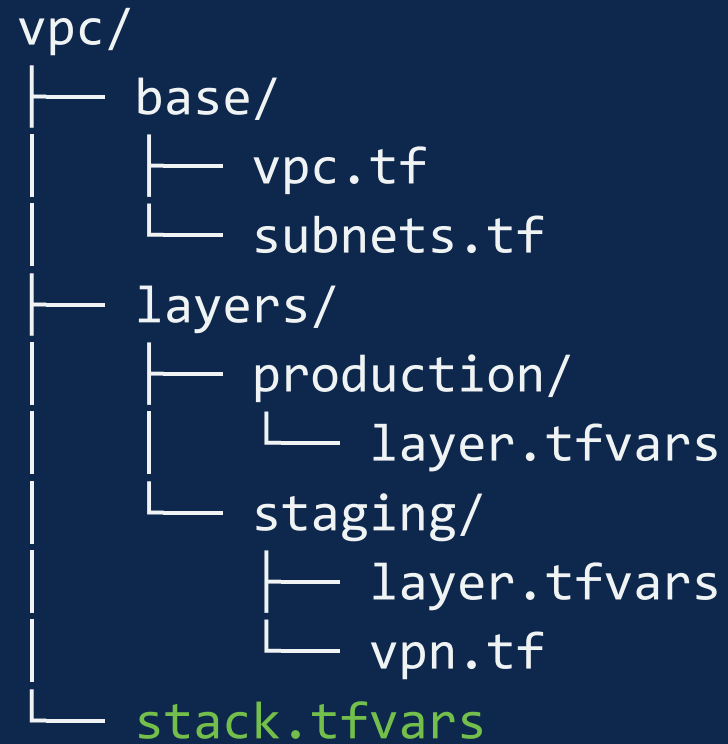
Example



Example

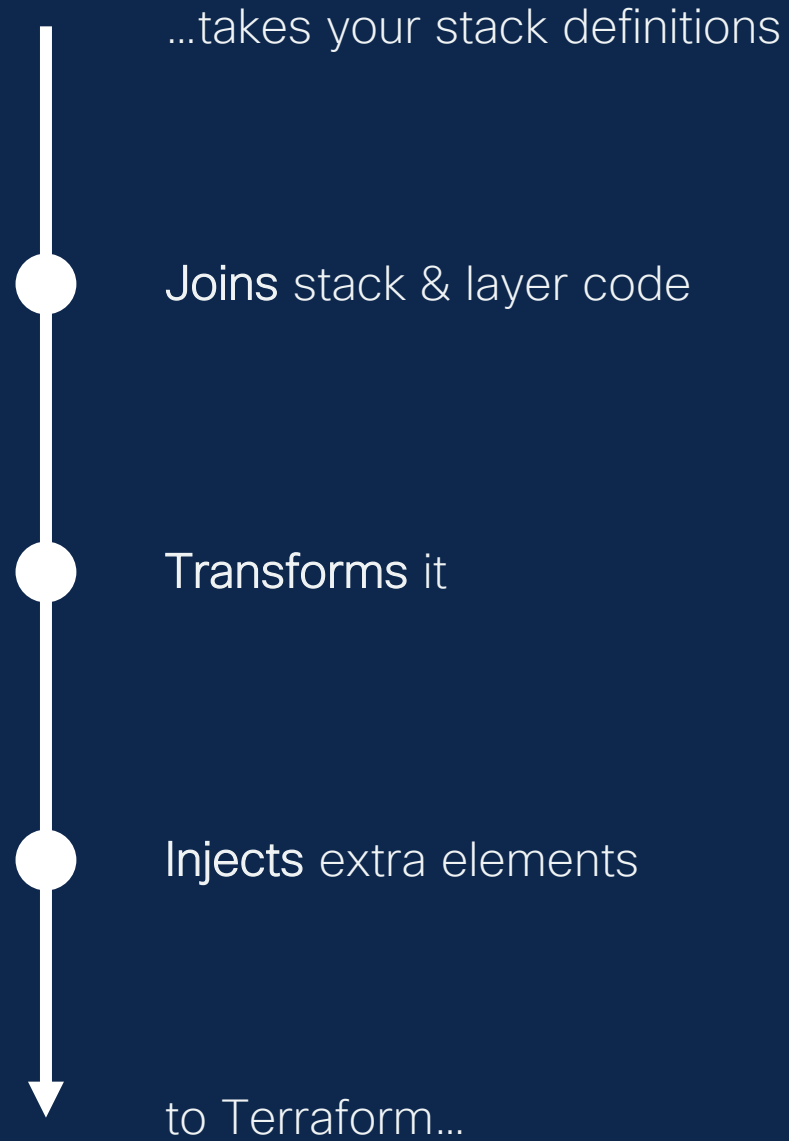


Example



Our solution

Workflow



Our solution

Plan times in check

We maintain deployment **separation**

Plan times remain **the same**



Our solution

Drift under control

Prevent drift by keeping it in sight

No need to limit **customizability**



Boilerplate gone

We inject boilerplate for you

Backends, providers, variables, etc.

Enforces **uniformity**



Our solution

Extra features

Cascading **variable scopes**

Auto-injected **state backend, providers, etc.**

Auto-declaring **variables**

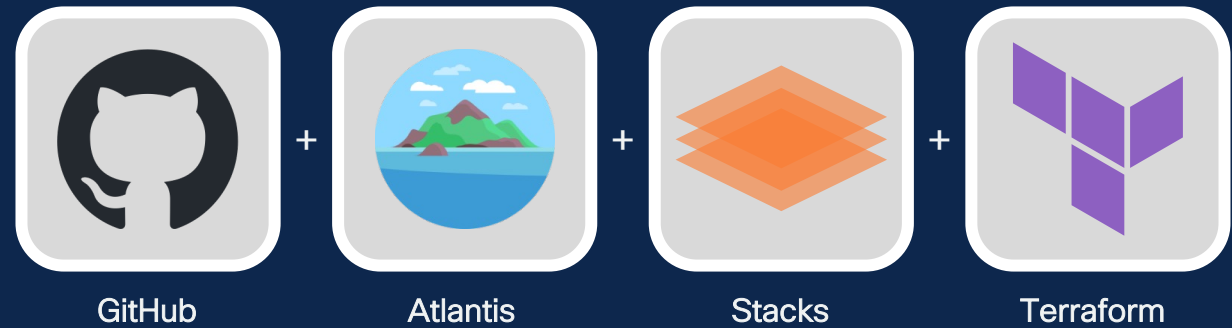
Secrets injection

Jinja2 templating

...and more



Our reference architecture



Key Takeaways

Terraform is still great

Modules only get you so far,
we need a better way to scale

Code preprocessing works

Stacks for Terraform



Our solution

It's open
source!



github.com/cisco-open/stacks



Q&A time!

Check us out!

thousandeyes.com

We are hiring!

thousandeyes.com/careers

