Scaling Terraform at ThousandEyes

SREcon23 Americas

Ricard Bejarano

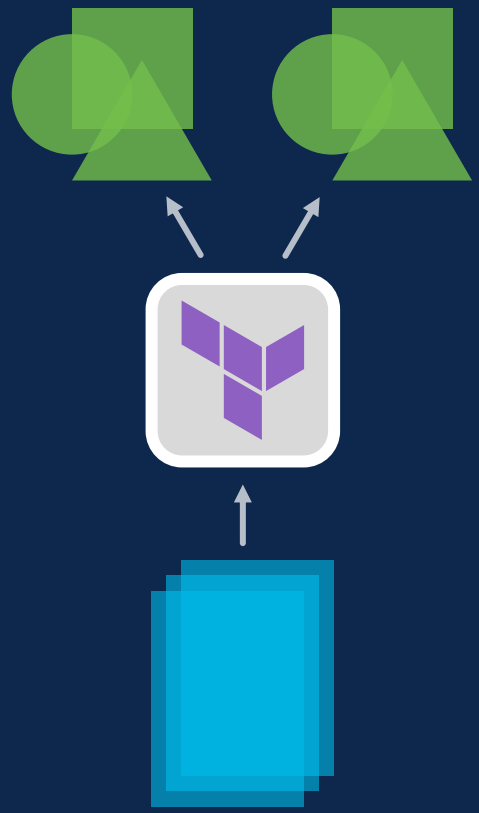
Site Reliability Engineer, Infrastructure

ThousandEyes
part of Cisco

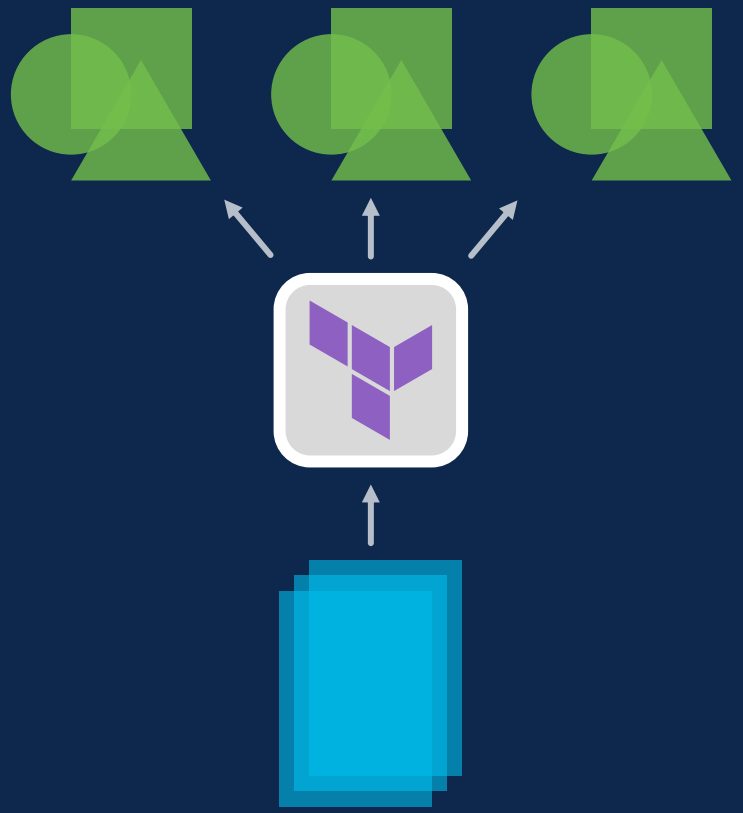
The problem

First Terraform deployment

Do that again



Scale to n



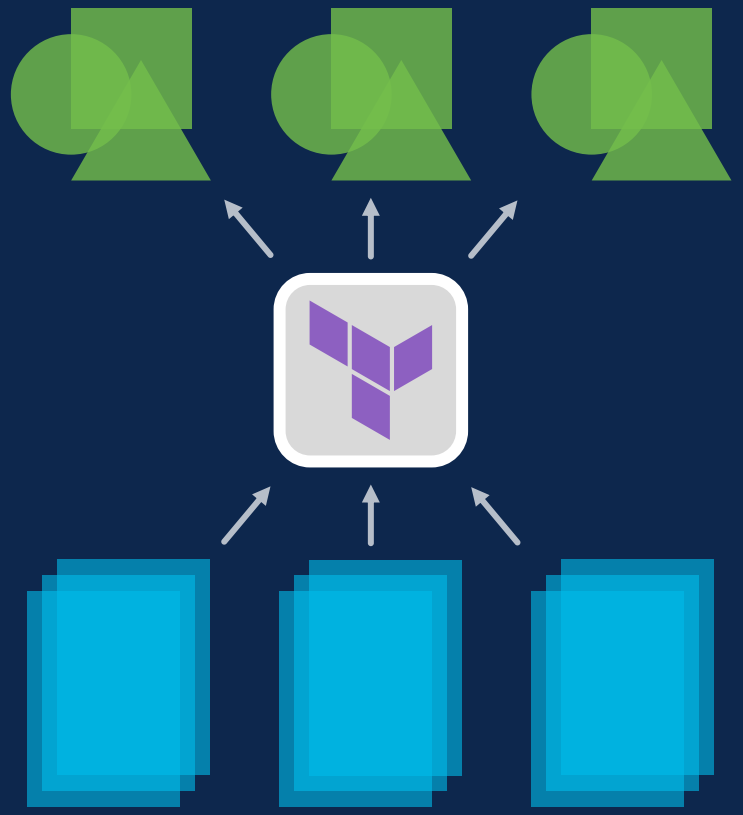
Plan **times**

Any **single change** triggers a plan

Terraform refreshes **every resource**

No, you **can’t really skip** refresh

Split into **separate** deployments



Drift potential

Separate deployments means **different files**

Differences across those files is **drift**

Reconciling drift in Terraform is **awful**

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 reinstantiate the same state backend, providers, variables, etc.

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

This doesn’t scale

ThousandEyes
part of Cisco

Our solution

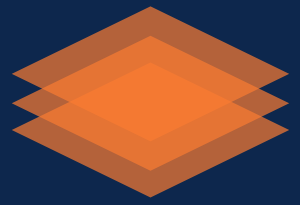
We looked into Terragrunt

We already had **5000+** Terraform files

Required **learning** a new tool

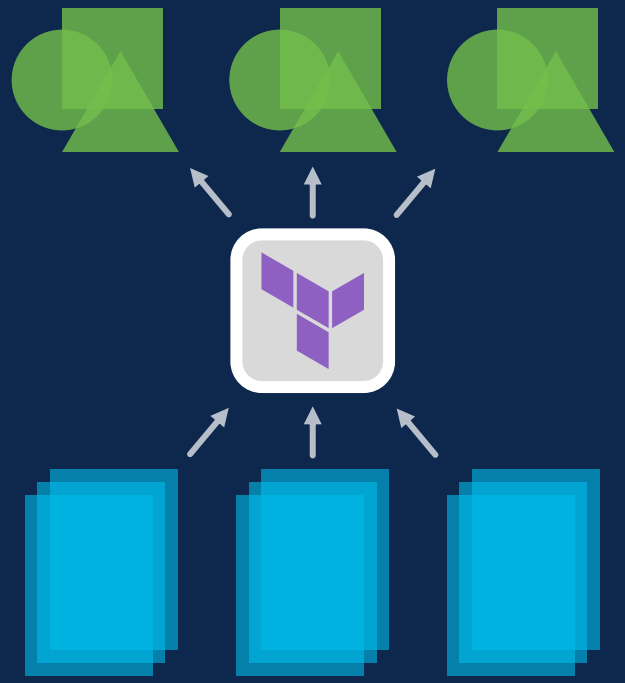
Didn’t solve the **boilerplate** problem

So we built our own

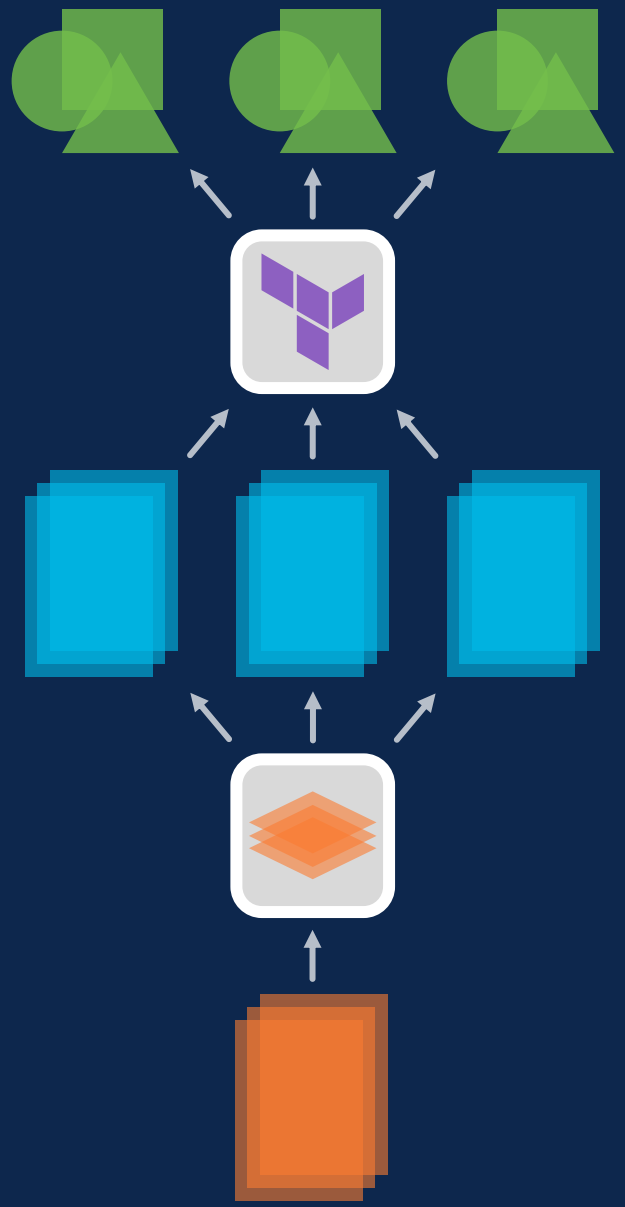


Stacks for Terraform

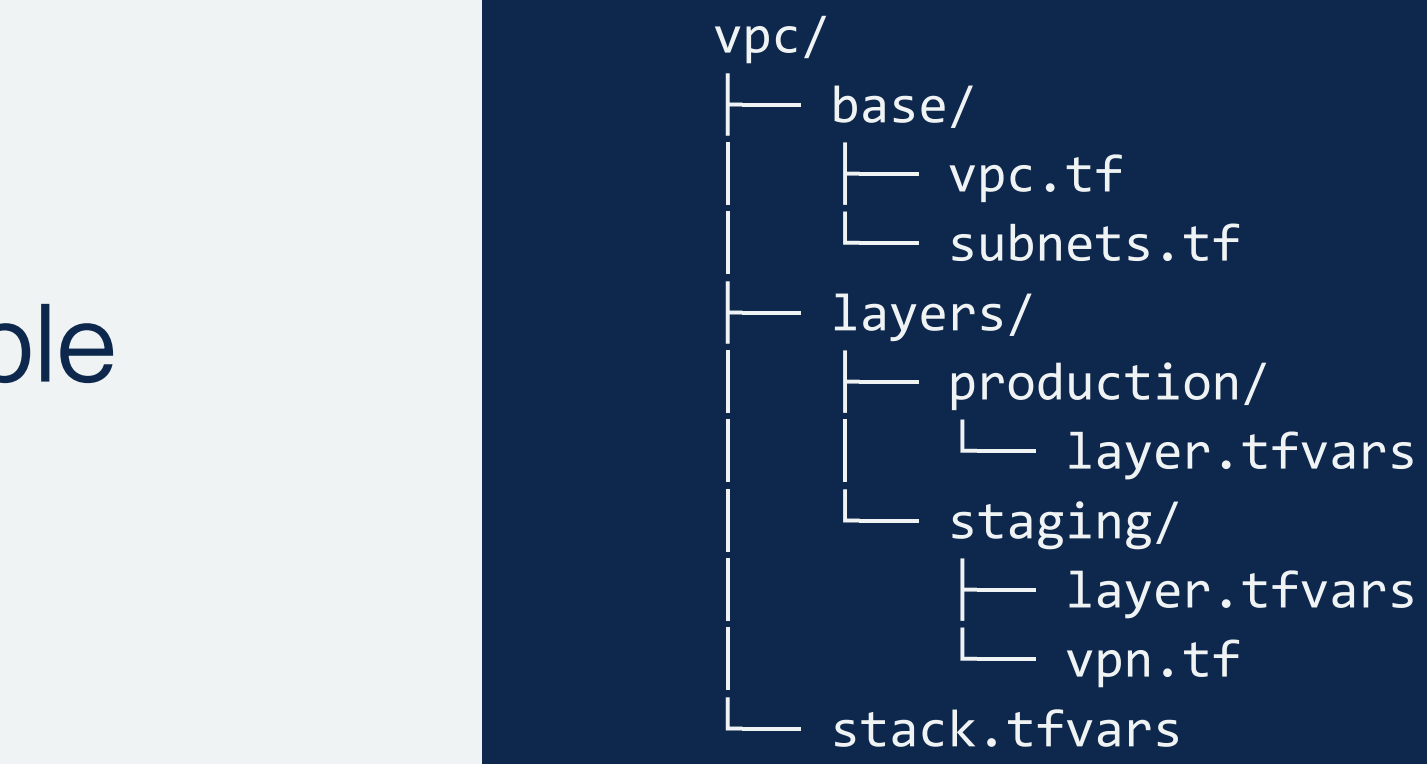
Remember this?



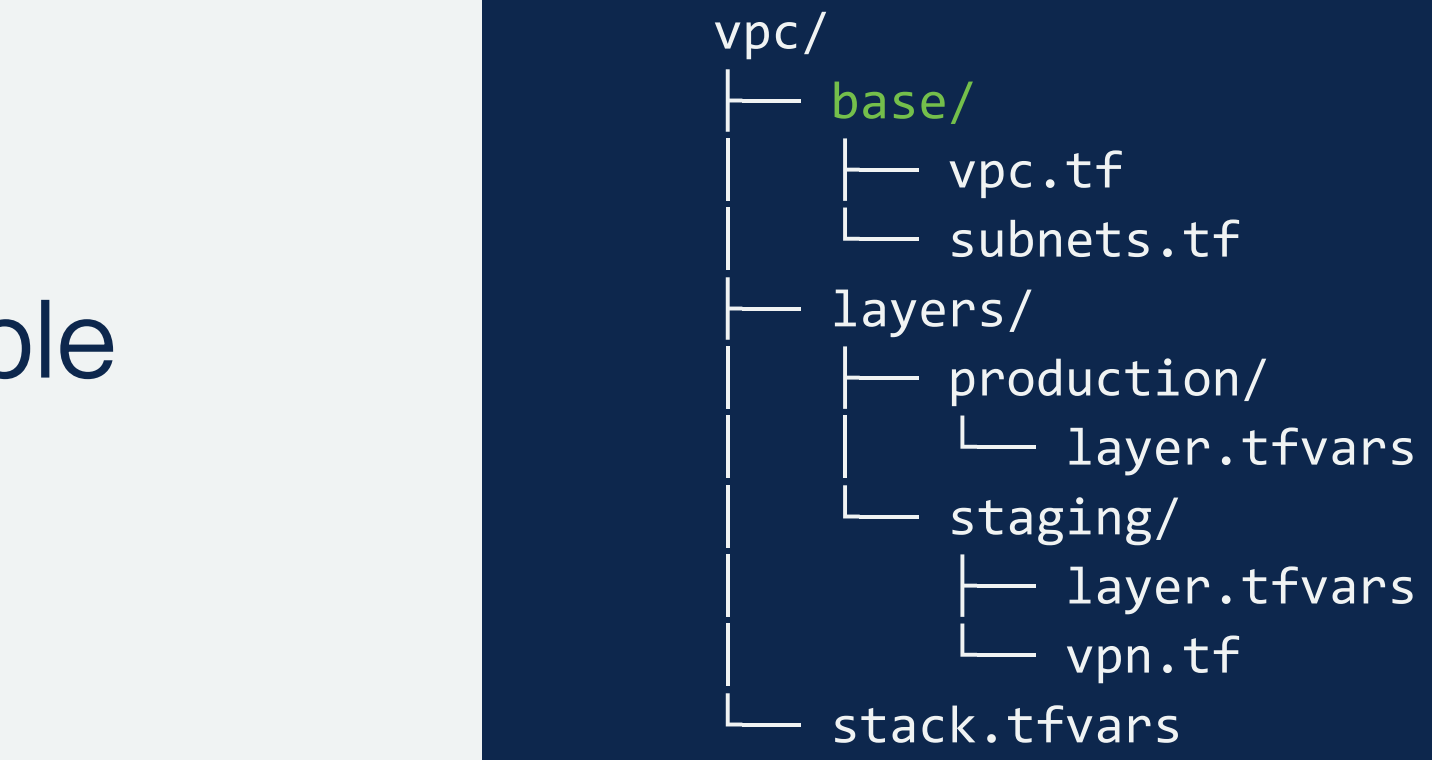
Stacks for Terraform



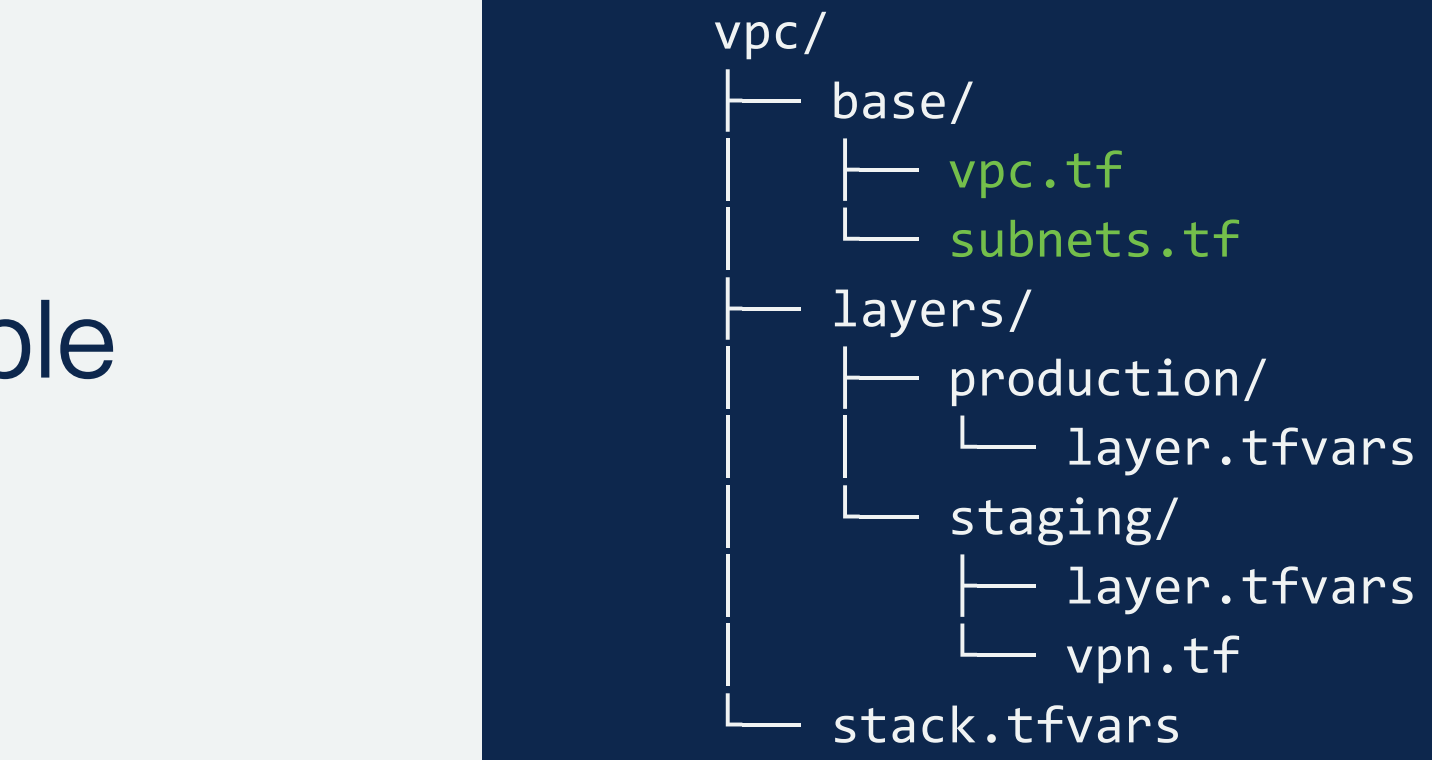
Example



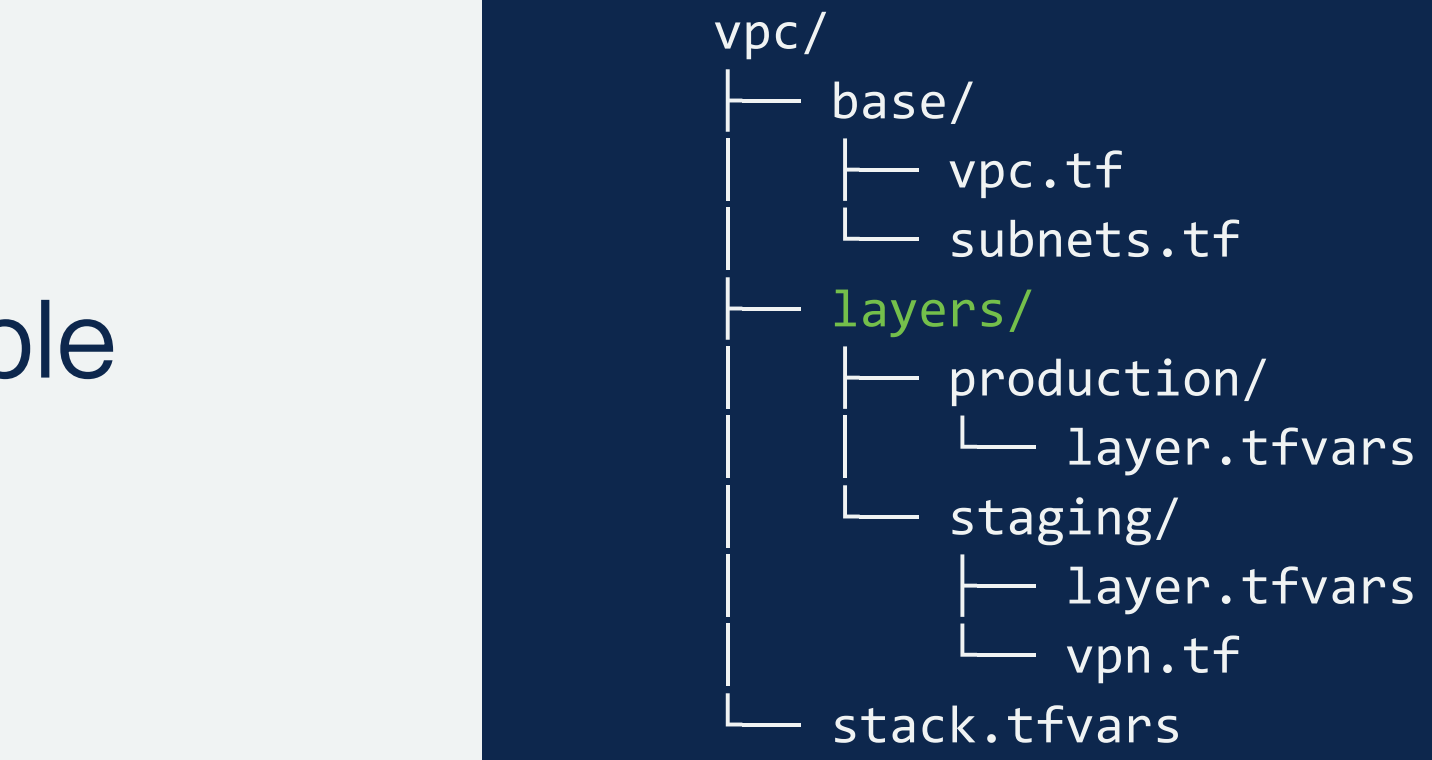
Example



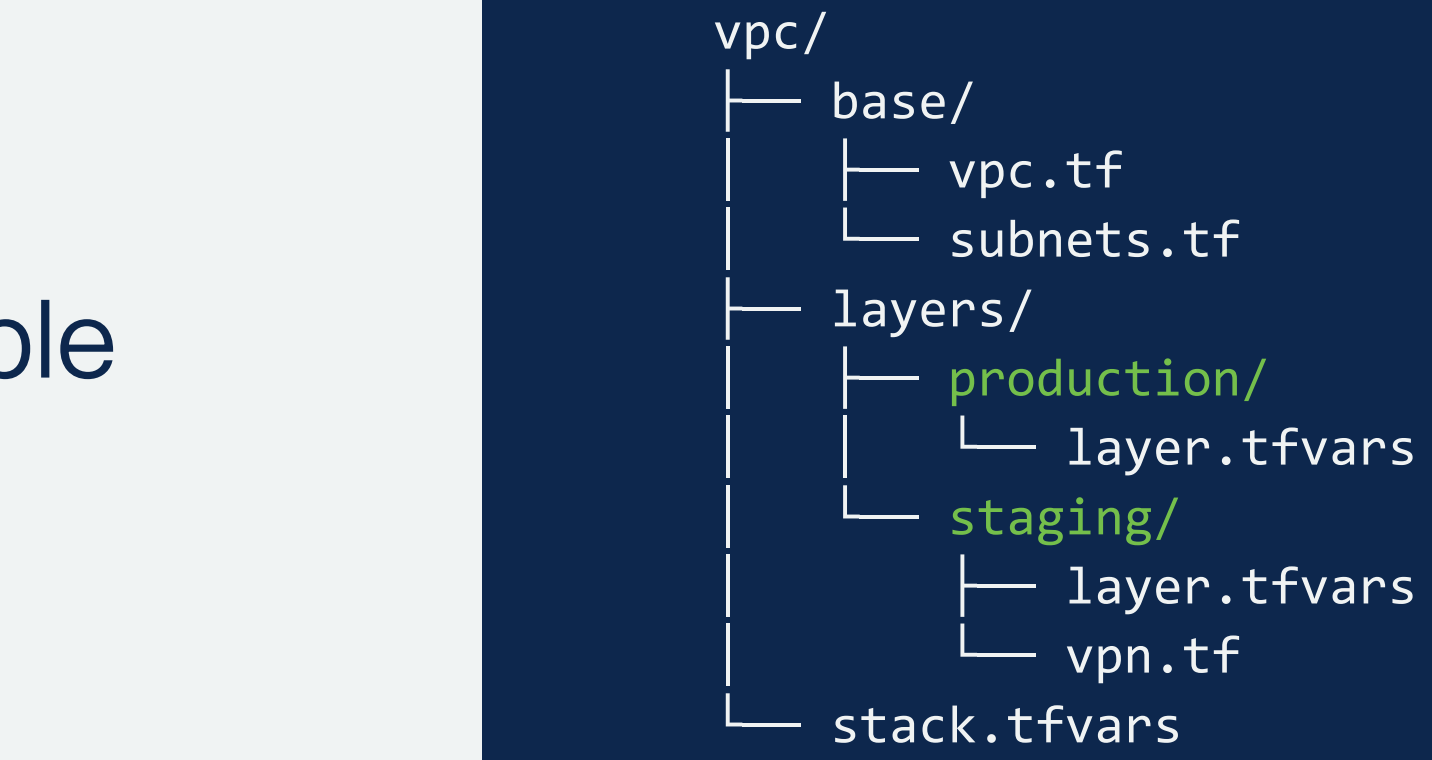
Example



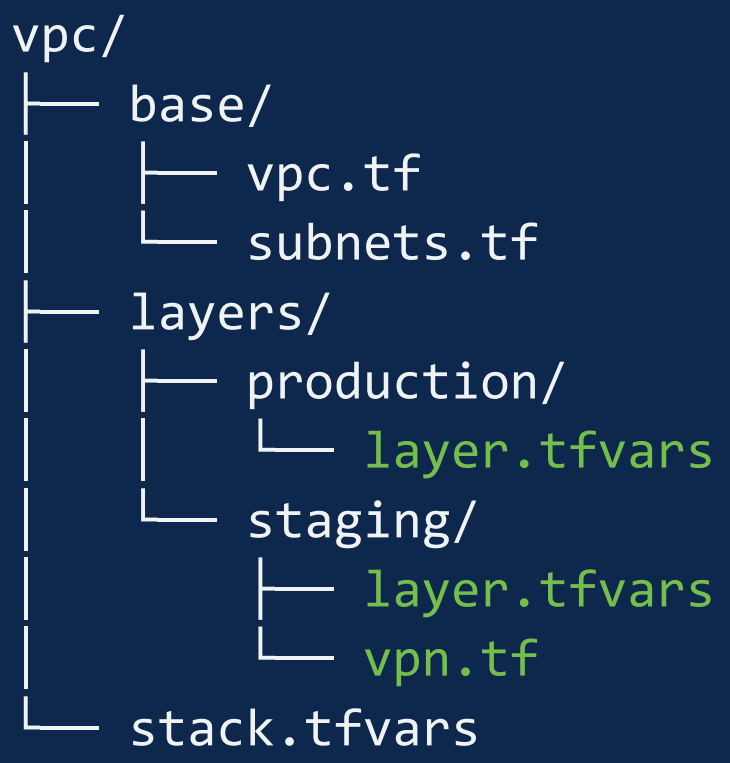
Example



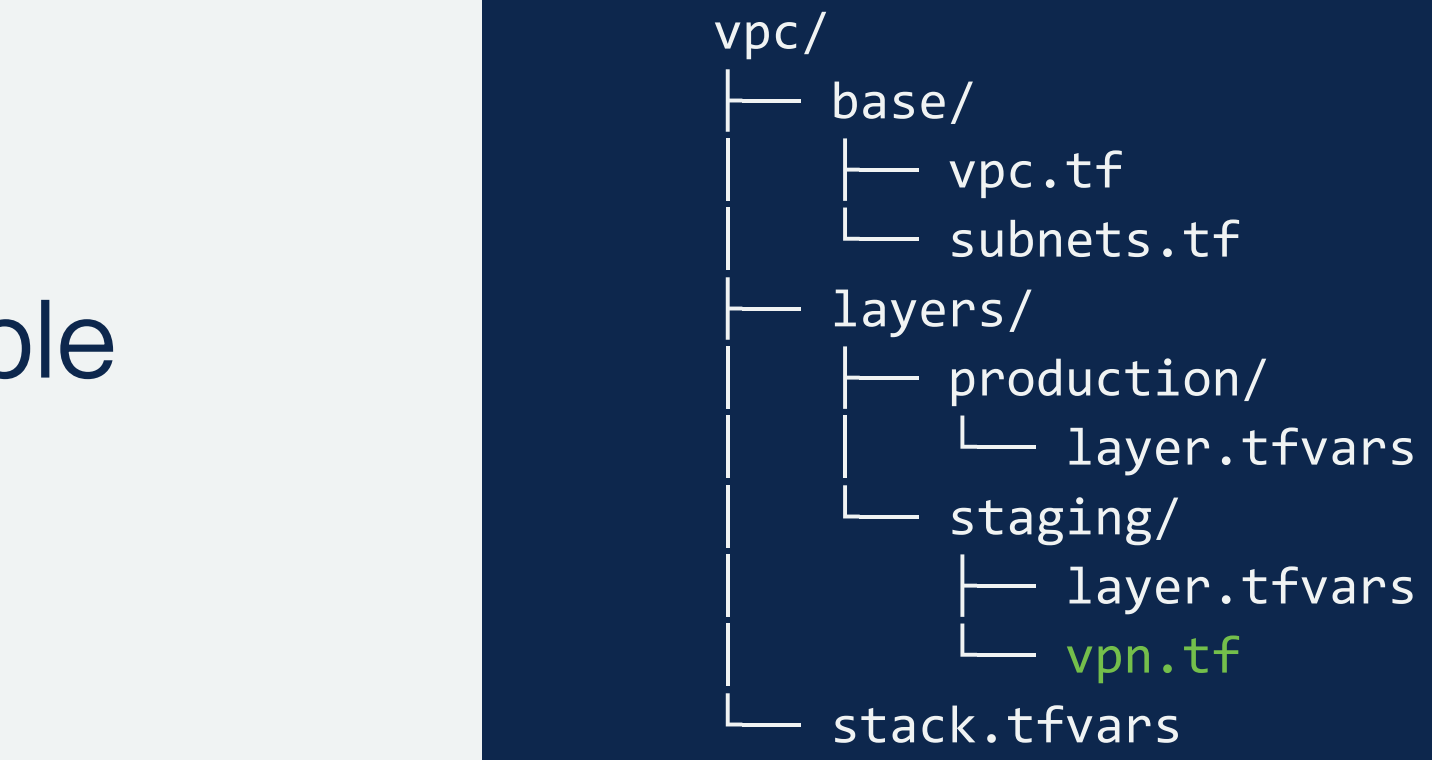
Example



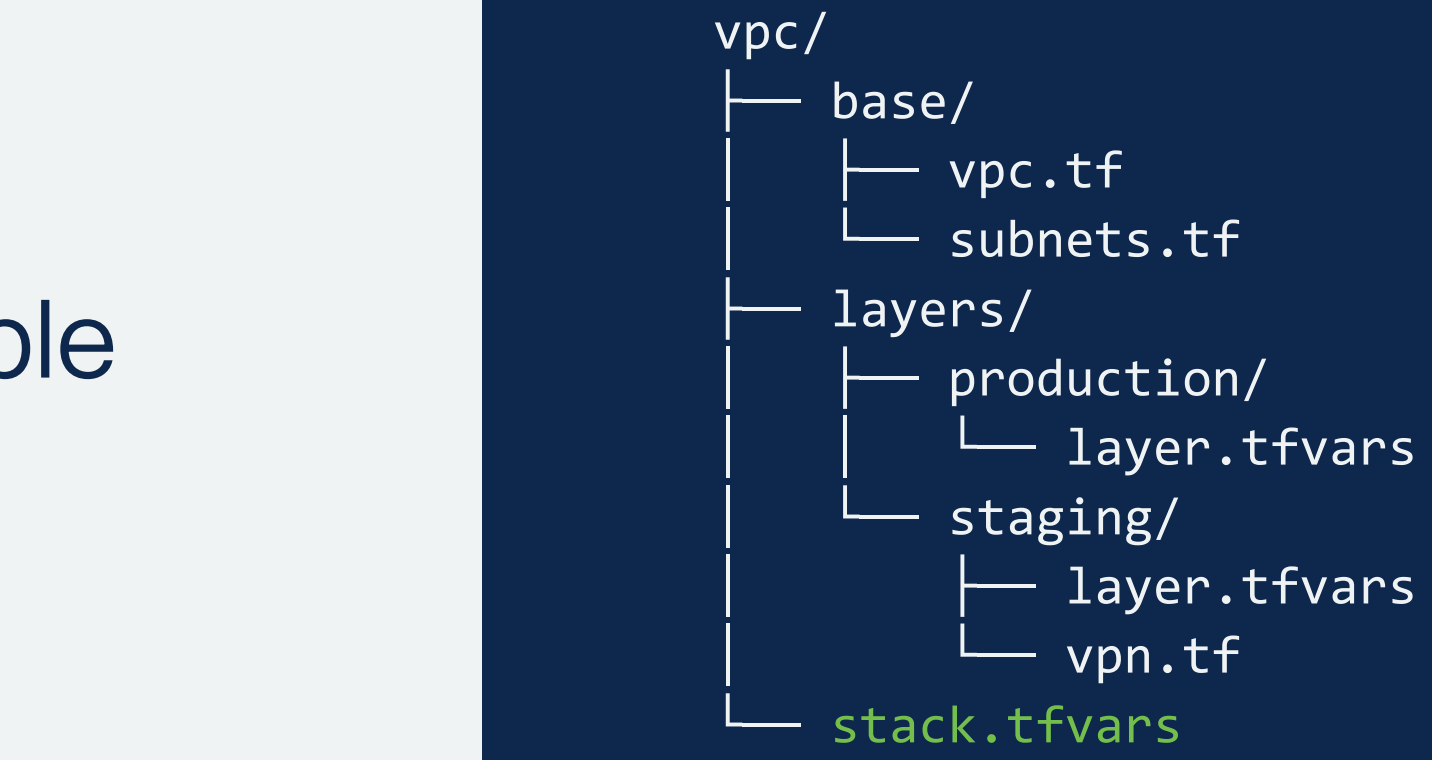
Example



Example



Example



Workflow

...takes your stack definitions

Joins stack & layer code

Transforms it

Injects extra elements

to Terraform...

Plan **times** in check

We maintain deployment **separation**

Plan times remain **the same**

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**

Extra features

Cascading **variable scopes**

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

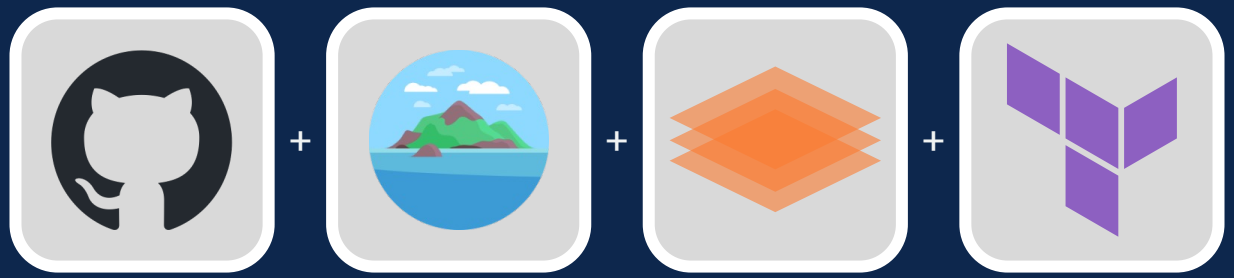
Auto-declaring **variables**

Secrets injection

Jinja2 templating

...and more

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

Open source!

github.com/cisco-open/stacks

ThousandEyes
part of Cisco

Q&A time!

ThousandEyes
part of Cisco

Check us out!

thousandeyes.com

ThousandEyes
part of Cisco

We are hiring!

thousandeyes.com/careers

ThousandEyes
part of Cisco

Talk to me!

ricard@bejarano.io